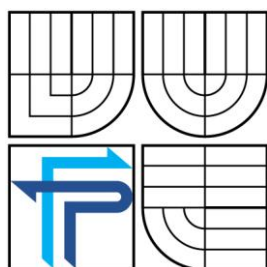


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ
ÚSTAV MANAGEMENTU**

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUT OF MANAGEMENT

ANALÝZA A NÁVRH INFORMAČNÍHO SYSTÉMU PRO SPRÁVU PODNIKU

ANALYSIS AND DESIGN OF COMPANY MANAGEMENT SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETR RENIERS

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR DYDOWICZ, Ph.D.

ZADÁNÍ DIPLOMOVÉ PRÁCE

Reniers Petr, Bc.

Řízení a ekonomika podniku (6208T097)

Ředitel ústavu Vám v souladu se zákonem c.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

Analýza a návrh informačního systému pro správu podniku

v anglickém jazyce:

Analysis and Design of Company Management System

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření "Licenční smlouvy" dle autorského zákona.

Seznam odborné literatury:

BASL, J. Podnikové informační systémy: podnik v informační společnosti. 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 2008. 283 s.: il., portréty. ISBN 978-80-247-2279-5.

KOCH, M. Management informačních systémů. vyd. 2., přeprac. Brno: Akademické nakladatelství CERM, 2008. 193 s.: il., grafy, tab. ISBN 978-80-214-3735-7.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. rozš. vyd. Praha: Ikar, 2000. 178 s.: il. ISBN 80-247-0087-5.

REPA, V. Analýza a návrh informačních systémů. 1.vyd. Praha: Ekopress, 1999. 403 s.: il. ISBN 80-86119-13-0.

VLASÁK, R. Základy projektování informačních systémů. 1. vyd. Praha: Karolinum, 2003. 144 s. ISBN 80-246-0727-1.

Vedoucí diplomové práce: Ing. Petr Dydowicz, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/2010.

L.S.

PhDr. Martina Rašticová, Ph.D. doc.
Ředitel ústavu

RNDr. Anna Putnová, Ph.D., MBA

V Brně, dne 26. 04. 2010

Abstrakt

S příchodem informačních technologií se začínaly objevovat nové typy webových aplikací a systémů, které byly vytvořené na standardních technologiích, a proto jsou dostupné na většině uživatelských počítačů. Bez těchto systémů si v dnešní době nedokážeme podniky a svět ani představit. Mezi podniky, uživateli a vývojáři si získali velkou oblibu a jsou vyvíjeny jak pro internetové aplikace, tak i pro firemní intranetové aplikace. Vzhledem k příchodu objektově orientovaného návrhu a návrhových vzorů je důležité si tyto moderní vývojové nástroje otestovat na vývoji informačního systému a analyzovat dnešní situaci na trhu s informačními systémy pro malé podniky. Právě touto problematikou se zabývá tato práce. Vyvíjím informační systém pro správu podniku, který slouží k lepší přehlednosti, správě firem, snížení nákladů a zvýšení zisku.

Klíčová slova

Analýza, návrh, SWOT, ERP, komunikace, informační systém, UML, návrhové vzory, MVC, objektová orientace

Abstract

With arrival of the information technology new types of easily accessible web applications and systems developed on standard technologies have begun to emerge. Nowadays we cannot even imagine the companies and the world without these systems. They have garnished massive popularity among companies, users and developers while being created for the internet applications as well as corporate intranet applications. Considering an emergence of object oriented design and design patterns it is important to test these modern tools on development of an information system and to analyze situation on the market with small company information systems. This thesis acts just this problem. I develop company management system, which serves for better management of firms, cost reduction and gain enhancement.

Keywords

Analysis, design, SWOT, ERP, communication, information system, UML, design patterns, MVC, object orientation

Citace

RENIERS, P. *Analýza a návrh informačního systému pro správu podniku*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2010. XY s. Vedoucí diplomové práce Ing. Petr Dydowicz, Ph.D.

Analýza a návrh informačního systému pro správu podniku

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petra Dydowicze, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Reniers
10.5. 2010

Poděkování

V této sekci bych chtěl poděkovat svým přátelům za vytrvalou podporu a v neposlední řadě Ing. Petru Dydowiczovi, Ph.D. za čas a ochotu k vedení této práce.

© Petr Reniers, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě podnikatelské. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	9
2 Vymezení problému a cíle práce.....	10
3 Teoretická východiska práce.....	12
3.1 Teoretické přínosy.....	13
3.2 Teoretická návratnost.....	13
4 Analýza problému a současné situace	14
4.1 Analýza zavedení nového informačního systému do firmy	16
4.1.1 Průzkum komunikace v podniku.....	17
4.1.2 SWOT analýza zavedení nového informačního systému do firmy	20
4.1.3 Současné podnikové informační a řídicí systémy	21
4.1.4 Možné alternativy řešení a jejich zhodnocení.....	24
4.1.5 HOS analýza vhodnosti nasazení systému do firmy	29
4.1.6 Závěr analýzy nasazení informačního systému do firmy	30
5 Vlastní návrh řešení	31
5.1 Analýza požadavků	31
5.1.1 Požadavky na informační systém.....	31
5.1.2 Role uživatelů	32
5.1.3 Požadavky na uživatelské rozhraní	33
5.1.4 Use Case diagram	33
5.2 Návrh informačního systému	35
5.2.1 Objektová orientace	35
5.2.2 UML	37
5.2.3 Softwarová architektura.....	37
5.2.4 Návrhové vzory	39
5.2.5 Architektura MVC.....	41
5.2.6 Sekvenční diagram	44
5.2.7 Diagram aktivit.....	46
5.3 Implementace	47
5.3.1 Technologie na straně klienta	48
5.3.2 Technologie na straně serveru	48
5.3.3 Použité prostředky	49
5.3.3.1 PHP.....	49
5.3.3.1.1 Výhody jazyka PHP	50
5.3.3.1.2 Nevýhody jazyka PHP	50
5.3.3.2 SQL.....	51

5.3.3.3 MySQL	51
5.3.3.1 Výhody jazyka MySQL	51
5.3.3.4 Webový server Apache	52
5.3.3.5 XHTML a CSS	52
5.3.3.6 JavaScript	53
5.3.4 Implementace uživatelského rozhraní.....	53
5.3.5.1 Vzhled.....	54
5.3.5 Ukázka implementace.....	56
5.3.6 Problémy při implementaci	58
5.3.6.1 Bezstavost HTTP	58
5.3.7 Náměty na rozšíření	59
5.3.7.1 Pokročilejší vyhledávání.....	59
5.3.7.2 Použití XML	59
5.3.7.3 Statistiky postupu práce	60
5.3.7.4 Rozšíření funkcionality systému o prvky ERP systémů	60
5.4 Ekonomické zhodnocení vlastního řešení	63
5.5 Porovnání uvažovaných variant	65
5.6 Struktura úkolů a činnosti řešení	67
5.7 Harmonogram a milníky řešení.....	68
6 Závěr	69
6.1 Hodnocení práce.....	69
Literatura.....	71
Seznam obrázků	73
Seznam tabulek	74
Seznam příloh.....	75

1 Úvod

Čas jsou peníze a to ví i majitelé a manažeři firem a společností. Účelem této práce je poskytnout malým firmám o desítkách uživatelů možnost správy podniku, které by finančně bylo přijatelné a časově ne příliš náročné a která by ušetřila každodenní schůzky a porady, které jsou mnohdy zbytečné, možnost prohlížení firemních informací pro vedoucí a manažery, lepší přehled o zaměstnancích apod. I ve firmě, která má stovky obyčejných zaměstnanců, vedoucích a manažerů, nic takového není a nastává mnohdy zmatek a chaos. Jsou zde komunikační mezery a to se dá vyřešit informačním systémem. Šetří to tedy čas i nervy a hlavně peníze.

Při vývoji informačního systému použiji objektově orientovaný "formální" návrh systému. Základem bude návrh, který bude postaven na softwarové architektuře Model View Controller.

Dále zmíním strukturu technické zprávy. Ve druhé kapitole vymezím problém a hlavní cíle práce. V následující kapitole se budu zabývat teoretickými východisky práce a ve čtvrté kapitole analýzou problému a současné situace. V páté kapitole jsou popsány konkrétní požadavky na informační systém, podle požadavků důkladně navrhnut tento informační systém a do detailů jsou rozebrány modelovací techniky UML. Pro návrh systému jsem použil různé typy diagramů. Přímě tyto diagramy jsou zobrazeny v této kapitole. V té samé kapitole jsou popsány technologie použité při vývoji. Pro implementaci této aplikace jsem zvolil kombinaci programovacího jazyka PHP a databázového systému MySQL. Nalezneme zde také i základní informace o ostatních použitých prostředcích. Dále kapitola se zabývá možnými nápady pro vylepšení a rozšíření a v neposlední řadě ekonomickým zhodnocení a dalšími alternativami V šesté, závěrečné kapitole se podíváme na zhodnocení práce a přínosu pro mě.

2 Vymezení problému a cíle práce

Jak je uvedeno ve zdroji [23], tak v současné době procházejí naše podniky celou řadou změn. Změny, které se týkají nasazování a využívání výpočetní techniky, patří k nejdůležitějším. Efektivní přístup k požadavkům zákazníka, k rychlému přijetí zakázky, včetně stanovení ceny a termínu, dále jejímu včasnému zajištění a dodání, nutí naše podniky k dalším krokům. Jde jednak o organizační změny, ale i o zavedení takového softwarového integrovaného informačního systému, který pracuje nad společnou sdílenou databází a komplexně podporuje zajištění celého obchodního případu od přijetí, přes realizaci až po jeho dodání včetně ekonomického vyhodnocení.

Pro minulé období je charakteristické:

- převaha osobních počítačů v našich firmách
- pocíťovaná nedostatečná výkonnost hardwarových prostředků
- potřeby organizací se změnily a staré počítačové systémy jsou již nevyhovující
- manažeři nebyli dříve vybaveni dostatečnými zkušenostmi a nedokázali přesně definovat funkční požadavky, které by odrážely novou realitu

Pro současné období je charakteristické:

- rychlé tempo a rozsah změn
- pouhý prodej zboží a služeb již nestačí a uživatelé vyžadují stálou podporu a servis ze strany dodavatelů

Na základě těchto změn je třeba změnit podnikovou strategii při budování informačních systémů.

Hlavním cílem této diplomové práce je tedy do malých podniků o desítkách uživatelů navrhnout a zavést nový informační systém pro správu podniku, který zlepší řízení informačních toků jak uvnitř podniku, tak i v jeho okolí. V podstatě cílem je takový menší ERP systém.

K jednotlivým částečným cílům patří navrhnout systém pomocí objektivě orientovaného návrhu a následně jej implementovat a provést ekonomické zhodnocení. Dalším cílem této práce je seznámit se s principy fungování firemních záležitostí a upevnit si znalosti s programováním v PHP a jiných jazycích. Mezi další cíle patří

analyzovat současnou situaci na trhu s informačními systémy a pomocí průzkumu také zjistit, jak jsou na tom podniky bez podobného informačního systému s komunikací. A v neposlední řadě navrhnout alternativy a zhodnotit je.

K metodám a postupům, které budu používat v této diplomové práci, patří:

- analýza na trhu informačních systémů
- metoda HOS
- metoda SWOT
- průzkum ve firmě v podobě dotazníků
- v návrhu postup od analýzy požadavků až po samotný návrh informačního systému
- ekonomické zhodnocení a přínos vlastního řešení

3 Teoretická východiska práce

Informační systém (IS) je systém pro sběr, udržování, zpracování a poskytování informací a dat.

S odkazem na zdroj [22], tak podnikový informační systém vytvářejí lidé, kteří prostřednictvím dostupných technologických prostředků a stanovené metodologie zpracovávají data a vytvářejí z nich informační a znalostní bázi organizace sloužící k řízení podnikových procesů, manažerskému rozhodování a správě podnikové agendy (Sodomka, 2006).

Uvedená definice srozumitelně vystihuje význam informačního systému pro dynamický rozvoj podniku a současně naznačuje, co tvoří podstatu informačního systému. Základním předpokladem existence podnikového informačního systému je vybudované technologické prostředí, skládající se z technického vybavení pro provoz informačního systému. Druhou, stejně důležitou součástí podnikového informačního systému je samotný produkt - informační systém. Třetí důležitou součástí podnikového informačního systému jsou lidé. Jednak uživatelé, kteří informační systém využívají ke své činnosti a jednak dodavatelé, kteří provádějí implementaci informačního systému a následně poskytují podporu v jeho činnosti. Podnikový informační systém jako celek zajišťuje funkcionalitu pro evidenci, uchování a vyhodnocení uložených informací k řízení podniku, případně informace pro hodnocení stavu podniku jak z pohledu realizovaných činností, tak z pohledu dynamického a strategického rozhodování vedení podniku, případně jeho majitelů. [22]

Funkce informačního systému a systému řízení se prolínají a vzájemně prostupují (Tvrdíková, 2000). Z uvedené citace lze jednoznačně odvodit následující konstatování. Systém řízení podniku je ovlivňován stavem a kvalitou informačního systému a je determinován kvalitou a stavem informační technologie v podniku. Pokud se budeme tímto konstatováním zabývat dále, dojdeme k závěru, že úroveň informační technologie je dána úrovní jednotlivých komponent, z nichž se skládá, a nedostatečná úroveň jedné z komponent informační technologie může zásadně ovlivnit kvalitu informační technologie jako celku. Proto, aby podnik udržoval úroveň informační technologie na požadované úrovni, stejně jako všech jejích souvisejících komponent, stanovuje informační strategii podniku. Ta je jednou ze součástí globální strategie podniku, předurčené strategickými vizemi a cíli podniku. [22]

Strategický význam informačního systému pro budoucnost firmy či instituce je významným rysem současnosti, proto je potřebné věnovat pozornost zvyšování účinnosti informačních systémů (Tvrdíková, 2000). Strategie obecně je pojem zahrnující systematický a vědomě řízený proces organizace lidské činnosti se stanoveným cílem a metodou, definující postup jeho dosažení. [22]

3.1 Teoretické přínosy

Mezi nejdůležitější přínosy, které můžou informační systémy ve správě podniku přinést, jsou v první řadě psychologické, sociologické a zvláště ekonomické v podobě snížení nákladů, zvýšením zisku apod.

3.2 Teoretická návratnost

Jak bylo zmíněno, existuje mnoho přínosů pro zadavatele ze zavedení informačního systému, ovšem pouze ten ekonomický lze jasně a přehledně převést do čísel. Na začátek lze říci, že úspěch samotného informačního systému není zcela zřejmý a je přímo úměrný úspěchu toho daného podniku.

4 Analýza problému a současné situace

Situaci na počátku 90. let je možno charakterizovat jako období velkých změn na světových trzích, tzn.

- otevření i dříve chráněných trhů
- celková globalizace trhu
- zostření konkurence na mezinárodních trzích
- rychlý vývoj techniky a nových technologií
- zkracování životních cyklů výrobků
- bouřlivý rozvoj a nasazování prostředků výpočetní a komunikační techniky

To ve svém důsledku přináší tlak i změnu celkové podnikové strategie např. reengineering různých podnikových procesů. Znamená to i myšlenkový vývoj z výrobní orientace na orientaci marketingovou. Mění se i postoj ke konkurenci, dodavatelům, ale především k zákazníkovi. Aktivita výrobců se zaměřují především na zabezpečení kvality a výrobků služeb, které poskytují. Tyto vnější vlivy jsou umocněny ještě naší vnitřní situací a to zejména:

- ztráta a rozpad původních trhů
- hledání nových odbytišť
- změny vlastnických vztahů
- postupnou tvorbou potřebné legislativy

Na tuto novou situaci musí reagovat i naše podniky a po vyřešení operativních úkolů souvisejících s jejich existencí začínají hledat cestu jak v nových podmínkách zabezpečit stabilní dlouhodobou prosperitu. Jedním z důsledků je, že výše uvedené změny přinesly i odlišný pohled a požadavky na dostupnost informací v podniku. Dnešní management vyžaduje k přijímání kvalifikovanějších rozhodnutí dostatek informací a ty mu dnešní prostředky informačních technologií mohou nabídnout. Není třeba zdůrazňovat, že informace se v poslední době staly důležitým výrobním faktorem.

O potřebě informací a tedy nového IS v podnicích jsme si řekli výše. Po fúzi společností může být managementem zvoleno bohužel velice nešťastné řešení, tedy

zachování provozu původních IS ačkoliv byly uvažovány i jiné možnosti (např. sjednotit všechny IS pod jeden z používaných, zakoupit zcela nový SW). Náklady spojené s provozem nevhodného systému se dají těžko vyčíslit přesně, nicméně existuje odhad těchto nákladů ve výši několika i statisíců Kč. Tvoří je všechny nadnormativní zásoby, špatná rozhodnutí na základě špatných údajů, práce nevytvářející žádnou přidanou hodnotu, několik zaměstnanců navíc, jejich technické zabezpečení apod.

Jakou tedy zvolit budoucí technickou platformu? Pro jednotlivé uživatele je současná platforma PC zcela vyhovující. Operační systém může být zvolen na základě požadavků informačního systému. Pokud bude IS požadovat po jednotlivých uživateli ke své funkci OS MS Windows, je vhodné zvolit verzi XP, v opačném případě je možné uvažovat o některém systému z rodiny UNIX/Linux.

U serverů je situace odlišná. Vzhledem k požadovanému výkonu není nutné ustupovat od současné architektury, bylo by však vhodné unifikovat operační systémy pod jeden standard. V tomto případě bychom ovšem hovořili o poměrně vysoké investici, doprovázené rizikem výpadku systému a/nebo celé počítačové sítě. Tuto změnu tedy bude nutné ještě uvážit, opět v návaznosti na zvolený IS.

Pro výběr nového IS byly definovány požadavky, který musí tento software splňovat. Některé musí platit pro celý systém, jiné jsou směřovány na jeho jednotlivé moduly. Pro přehlednost byly jednotlivé požadavky rozděleny do kategorií dle platnosti.

Očekávaný vývoj vychází z předpokladu, že budoucí podnik bude mít plošší organizační strukturu a síťové propojení uvnitř podniku i s vnějšími zdroji informací. Z hlediska technického provedení je patrná orientace na objektově orientovanou výstavbu, aplikování client-sever technologií, důsledné využívání možností CASE pro udržení konzistentního vývoje a zabudování změn do výstavby IS. Snaha využívat i pro nové verze stávajícího hardware a software a lze provozovat jak textově orientované pracovní terminály, tak i grafické osobní počítače. Softwarové firmy již pracují na změně svých produktů do prostředí Windows.

Z hlediska funkčních změn je nutné v modulech informačního systému reagovat na změnu filosofie od orientace na výrobu k orientaci na výrobek a zákazníka, tzn. přesun těžiště funkcí od řízení výroby k řízení zakázek. Proto součástí IS podniku budou muset být nástroje pro projektové řízení a řízení zakázek, podpora pro zpracování nabídek, přijímání zakázek, jejich termínování, plánování, vč. sledování ekonomické a finanční stránky tohoto procesu, kalkulace a výsledný controlling.

Informační systém by měl umožňovat tvořit samostatná, později rozšiřitelná řešení, která postupně pokryjí potřebné funkce v podniku. Svou pružností musí zabezpečit i podporu těch požadavků podniku, kterými bude reagovat na své téměř trvale se měnící potřeby v budoucnu.

4.1 Analýza zavedení nového informačního systému do firmy

Mnou vybraný podnik pro zavedení nového informačního systému je firma Unicode systems, s.r.o. Takže nejdříve k popisu této firmy.

Společnost Unicode Systems, s.r.o. je na našem trhu dostatečně známá svými informačními systémy pro čerpací stanice. V poslední době zaznamenává úspěchy nejen na našem trhu, ale podniká první úspěšné kroky na trhu v Rakousku, Maďarsku a dalších zemích. Jádrem nynějšího týmu pracovníků Unicode začalo s vývojem a výrobou automatizovaného výdeje nafty už v roce 1988. Tehdy vše probíhalo pod hlavičkou jednoho prosperujícího JZD na jižní Moravě. Firemní řešení – jako ostatně mnohá řešení těch dob – bylo typu „jednooký mezi slepci“. Zkrátka využilo se nedostatku podobných zařízení na našem trhu. Šlo to tak daleko, že se vyvinul vlastní snímač karet s čárovým kódem a podobné pionýrské náhražky nedostupné techniky ze Západu. Hned po revoluci byla založena soukromá firma o dvou osobách. První zákazník byly v roce 1990 Dopravní podniky Praha, kde je Unicode Systems stálým dodavatelem tankovacích automatů dodnes. První zlom nastal v roce 1993, kdy jsme vyhráli výběrové řízení pro Policii České republiky. Tehdy Unicode nainstaloval kolem 40 tankovacích automatů na policejních čerpacích stanicích. Následující rok jsme zpracovali centralizaci dat s přenosem do centrály Ministerstva vnitra ČR. Druhý významný zlom byl náš vstup na trh veřejných čerpacích stanic v roce 1996. Unicode společně s PETROLMEDIA s.r.o. nabízí stále se rozvíjející systém Shop 2000 Central. Z posledních roků firma Unicode Systems uzavřela smlouvu na dodávku informačního systému EuroShop na všech 1200 čerpacích stanic OMV clusteru 1. Systémem Unicode EuroShop tak bude během následujících 3 let vybaveno 520 stanic v Rakousku, 170 stanic v Maďarsku a 410 stanic v Německu. Když si to shrneme stručně, tak společnost se zabývá touto činností: řídicí a informační systémy pro čerpací stanice; samoobslužné výdejní terminály pro výdej pohonných hmot; pokladní systémy pro veřejné čerpací stanice; pokladní systémy pro neveřejné čerpací stanice; centrální zpracování dat sítě čerpacích stanic; centrální

zpracování dat prostřednictvím internetu. Má 65 zaměstnanců. Například vývojový tým má 19. Takže je celkem malá firma o desítkách zaměstnanců a je vhodná pro implementaci mého řešení informačního systému pro správu podniku.

4.1.1 Průzkum komunikace v podniku

Tato kapitola se zabývá průzkumu způsobu komunikace v malé firmě. Každý manažer by se měl v první řadě soustředit na dostatečnou úroveň informovanosti svých zaměstnanců např. o firemních cílech, záměrech a perspektivách dalšího rozvoje. Problémem však v tomhle případě podle mého názoru je, že často manažeři sami toto nejsou schopni srozumitelně formulovat a tudíž nemohou tyto poznatky předávat svým podřízeným. Každý vedoucí pracovník by se měl zamyslet nad tím, jakým způsobem svým podřízeným sděluje zásadní informace o dění v podniku, firemních cílech a záměrech. Pokud se podíváme na aktuální trendy ve vyspělých zemích, zjistíme, že cílem je právě větší míra spoluúčasti, zainteresovanosti a seberozvoje ze strany samotných zaměstnanců. O téhle problematice komunikace v malé firmě pojednává práce dle zdroje [25], ze které částečně čerpám.

Interní komunikace je základem celého fungování firmy, vychází z firemní kultury, z chování a z jednání managementu firmy a z komunikačních znalostí a dovedností jednotlivých manažerů, ze správného nastavení informační a komunikační infrastruktury firmy. Systém firemní komunikace je jedním ze základních nástrojů řízení a uplatňování moci (vlivu) ve firmě, který ovlivňuje firemní procesy a jednání lidí uvnitř i vně firmy a je také jedním ze základních prostředků pro vytváření, rozvíjení (řízení) a případnou změnu firemní kultury [25].

První dílčí výzkumná otázka zní: „Jak jsou zaměstnanci informováni o firemních hodnotách a cílech?“. Na tuto otázku se pokusím odpovědět s pomocí odpovědí zaměstnanců z dotazníku. Z celkového počtu 25 dotazovaných 14 zaměstnanců vnímá svoji informovanost o cílech firmy jako dostatečnou (56%). Naopak 11 zaměstnanců nesouhlasí s tím, že mají dostatek informací o firemních cílech. S informovaností o cílech na úrovni týmu je to mnohem pozitivnější, celkem 22 zaměstnanců (88%) se domnívá, že jsou dostatečně informováni a jen 3 zaměstnanci (12%) s tímto tvrzením nesouhlasí. S tvrzením, že mají dostatek informací o hodnotách firmy, souhlasí 15 zaměstnanců (60%), naopak 10 zaměstnanců (40%) s tímto tvrzením nesouhlasí [25].

Tabulka č. 4.1 - Informovanost o hodnotách a cílech

	ano, naprosto souhlasím		spíše souhlasím		spíše nesouhlasím		ne, rozhodně nesouhlasím	
Mám dostatek informací o cílech firmy (o tom, čeho chce firma dosáhnout)	2	8%	12	48%	11	44%	0	0%
Mám dostatek informací o cílech mého týmu (o tom, čeho chce dosáhnout tým, jehož jsem součástí)	11	44%	11	44%	3	12%	0	0%
Mám dostatek informací o hodnotách firmy (o tom, co je v Tribunu pokládáno za důležité)	3	12%	12	48%	8	32%	2	8%

Dle průzkumu provedeného mezi zaměstnanci se nejčastěji o cílech dozvídají přímo od nařízeného (odpovědělo tak 7 zaměstnanců, tj. 28%) či neoficiálně (zprostředkovaně od kolegů – odpovědělo 12 zaměstnanců tj. 48%). Mezi další používané nástroje patří webové stránky firmy, kde firemní cíle zjistilo 2 zaměstnanci (8%), 3 zaměstnanci se domnívají, že nebyli nijak informováni (12%) a jeden zaměstnanec (4%) zvolil variantu vlastní – o cílech se dozvěděl postupem času. Celkem 10 zaměstnanců (40%) se domnívá, že o firemních hodnotách nebyli informováni. 8 zaměstnanců (32%) o firemních hodnotách informovali neoficiálně jejich kolegové. Od nadřízeného se o firemních hodnotách dozvěděli 3 zaměstnanci (12%), stejný počet zaměstnanců se o nich dozvěděl z webových stránek firmy, postupem času se o hodnotách firmy dozvěděl 1 zaměstnanec (4%) [25].

Tabulka č. 4.2 - Nástroje používané k informování o hodnotách a cílech

	od nadřízeného		z nástěnky		neoficiální cestou (zprostředkovaně od kolegů)		elektronickou poštou		z webových stránek firmy		nebyl jsem informován		jinak (postupem časů)	
O cílech firmy jsem se dozvěděl	7	28%	0	0%	12	48%	0	0%	2	8%	3	12%	1	4%
O firemních hodnotách jsem se dozvěděl	3	12%	0	0%	8	32%	0	0%	3	12%	10	40%	1	4%

Před zodpovězením této dílčí výzkumné otázky bych se rád pozastavil nad zjištěními plynoucími z odpovědí zaměstnanců. Zaměstnanci cítí svoji informovanost o firemních cílech jako dostatečnou jen v 56%, o firemních hodnotách se cítí být dostatečně informováno 60% z nich. O cílech se při tom největší část z nich dozvěděla neoficiální cestou – zprostředkovaně od kolegů (48%). Teprve za touto "šuškanou" se objevila oficiální cesta – od nadřízeného se o cílech firmy dozvědělo 28% zaměstnanců. Stejných 40% zaměstnanců, kteří odpověděli, že nejsou dostatečně informováni o firemních hodnotách, se později v další otázce vyjádřili, že o hodnotách nebyli informováni vůbec. Když už se zaměstnanci o firemních hodnotách dozvěděli, tak to bylo neoficiální cestou – zprostředkovaně o kolegu (32%). Firemní hodnoty svých podřízeným manažeři příliš nesdělují – pouhých 12% se vyjádřilo, že by tomu tak nebylo. Z výše zmíněného usuzuji, že všichni manažeři vnímají hodnoty a cíle spíše jen svého oddělení či týmu a už nevnímají nějaké firemní cíle a hodnoty. Z toho plyne i návrh na zlepšení. Manažeři by se měli při společné schůzi ujednotit ve vnímání firemních hodnot a cílu, tak aby je mohli předávat všem svým podřízeným. Podle odborníku (Bělohlávek, 2000, s. 72) by manažer měl společně s pracovníkem určit dílčí cíle vedoucí k dosažení cíle konečného. Je-li pracovník zaangažován do určení svých cílů, dá se očekávat, že bude podstatně více motivován k jejich dosažení. Vysoká míra informovanosti pracovníku je nezbytnou podmínkou konkurenceschopnosti podniku. Je tomu tak proto, že zaručuje funkčnost celého systému, zaručuje efektivní chod, zvyšuje motivaci pracovníku, umožňuje zpětnou vazbu každému jednotlivci a je předpokladem

pro rozvoj a inovaci. Umožňuje utváření a změny postojů, které jsou předpokladem měnící se a učící se organizace (Mikuláščík, 2003, s. 214) [25].

4.1.2 SWOT analýza zavedení nového informačního systému do firmy

Ze SWOT analýzy vyplývá, že silnou stránkou zavedení informačního systému do podniku je především snížení celkových nákladů, zvýšení zisku, zrychlení vnitropodnikové komunikace a zpřehlednění vnitropodnikových dat. Stejně tak to může vést například i ke zlepšení služeb zákazníkům a k větší motivaci po zaměstnance.

Dále je zřejmé, že společnost, která zavádí nový informační systém, může hrozit z této činnosti například to, že zaváděcí proces bude zdoluhavý a nebudou ve firmě dostatečné znalosti a odborníci. Taktéž hrozbou může být nedostatek financí pro pokrytí jak zavedení informačního systému, tak i pokrytí nákladů z důvodu zpoždění výroby během nasazování informačního systému do podniku.

Příležitosti lze identifikovat zejména nových technologií a nových vnitropodnikových procesů, které by mohli znamenat zlepšení fungování podniku.

Nutnost nového informačního systému pro podniky je očividná. Příležitosti převažují nad hrozbami. Existují nyní dvě varianty řešení – je možné zvolit buďto informační systém na míru nebo vybrat některý z již existujících produktů na trhu (včetně případné úpravy dle požadavků podniku).

První řešení, tedy IS vytvořený na zakázku, má výhodu v dokonalém vyhovění požadavkům zákazníka. Takový návrh bude navrhnut v dalších kapitolách i spolu s alternativami tohoto řešení.

SWOT		Interní analýza	
		Silné stránky	Slabé stránky
Externí ananas	Příležitosti	snížení celkových nákladů zvýšení zisku zrychlení vnitropodnikové komunikace zpřehlednění vnitropodnikových dat snížení skladových zásob zlepšení služeb zákazníkům větší motivace zaměstnanců	nová technologie, která změní vnitropodnikové procesy potřeba zaškolení pracovníků potřeba nové techniky finanční náročnost časová náročnost
	Hrozby	nová technologie a vnitropodnikové procesy potřeba zaškolení pracovníků potřeba nové techniky finanční náročnost časová náročnost	zdlouhavý proces nedostatečné know how nedostatek financí na dokončení implementace nedostatek odborníků zpoždění výroby během implementace

Tabulka č. 4.3 - SWOT analýza výběru nového řešení provozu podnikového IS

4.1.3 Současné podnikové informační a řídicí systémy

Jak jsem řekl, tak navrhuji takový malý ERP systém pro malé podniky. V této kapitole se věnuji právě těmto systémům řízení podniků dle zdroje [24].

Uplynulé desetiletí bylo v podnicích ve znamení zavádění komplexních informačních systémů kategorie ERP (Enterprise Resource Planning). Bez nadsázky lze současné aplikace ERP považovat za ty, které nejvýrazněji ovlivňují řízení podniků. To dokazuje nejen počet implementací, ale především jejich dopad.

Charakteristika současných ERP

Systémy ERP dnes využívá více než 90% podniků zařazených u nás do žebříčku nejvýznamnějších firem Top100. Z toho je zřejmé, že ERP ovlivňuje rozhodování v podnicích s významným podílem na exportu, zaměstnanosti a tvorbě hrubého domácího produktu [24].

Na rozdíl od počátku 90. let si dnes dodavatelé ERP se svými zákazníky více rozumějí a vědí, co od sebe vzájemně mohou očekávat. A to i přesto, že tuto oblast dosud neopouští její vnitřní dynamika. Změny, ke kterým v těchto systémech dochází, lze ilustrovat na významu, jaký je přikládán písmenům zkratky ERP. Po původní dominanci písmena P (planning), vystřídané následně důrazem na R (resources), se nyní nejdůležitějším stává písmeno E, tedy podnik (enterprise). A to v širším slova smyslu, protože se jedná o podniky otevřené. To se mimo jiné projevuje rostoucím zprostředkováním přístupu k informacím pro partnery, dodavatele a zejména zákazníky podniku. Součástí řešení ERP se stávají i samostatně použitelné aplikace typu SCM, CRM nebo BI [24].

Příčiny neefektivního nasazení ERP

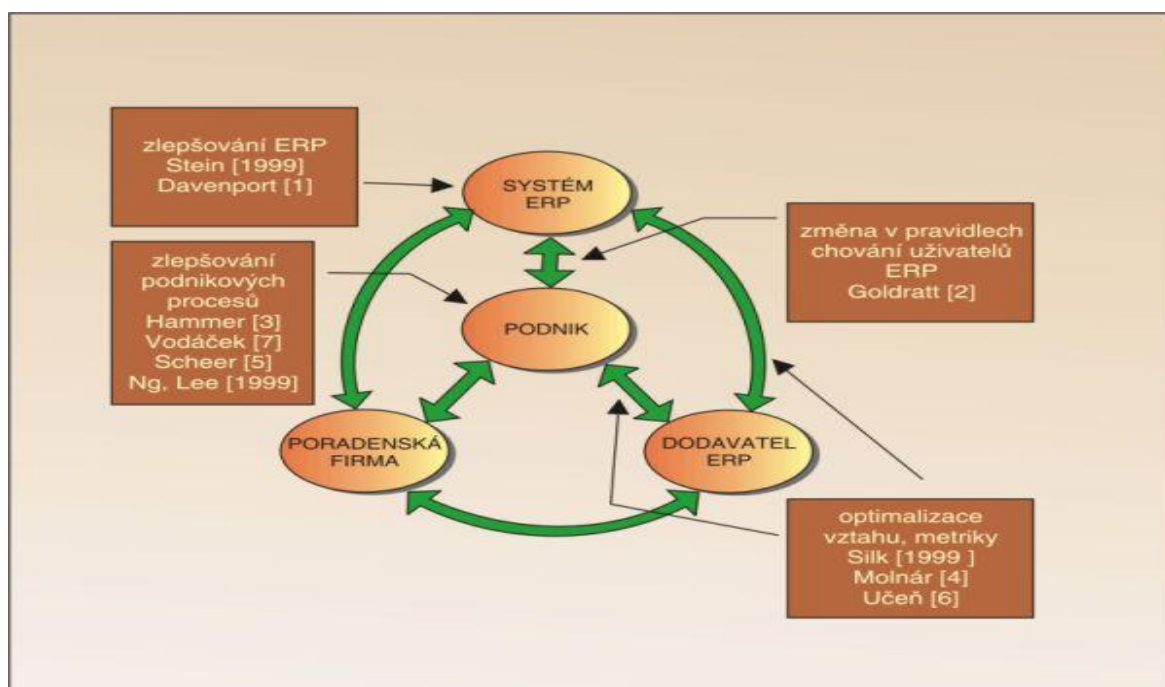
Záhy po vstupu ERP na trh počátkem 90. let se začaly objevovat první reakce na implementovaná řešení, které vždy nesplnila všechna očekávání. Ze schématu na obrázku č. 4.1 je patrné, že na správnou implementaci a provozování ERP nemá vliv jen vlastní produkt ERP a jeho dodavatel, ale i uživatel a v mnoha případech také externí poradenská firma. Neméně významné jsou i jejich vzájemné vazby. Obecně lze říci, že příčinami neefektivního nasazení ERP mohou být:

- nevhodný systém ERP
- nevhodný podnikový proces
- nesprávně fungující vztahy mezi dodavatelem a uživatelem, včetně metrik a smluv
- nevhodná pravidla chování uživatelů ERP

Schéma na obrázku č. 4.1 zachycuje významné autory poukazující na tyto problémy s uvedením roku zveřejnění a odkazem na literaturu. Vzhledem k určitému nasycení trhu a stabilizaci nabídky by se mohlo zdát, že ERP představují relativně ustálenou a

uzavřenou oblast. Opak je pravdou. V oblasti ERP dochází ke změnám (a mnohdy zcela zásadním), a to na nejrůznějších úrovních [24].

Obrázek č. 4.1 – Vlivy na správnou implementaci a provozování ERP



Trendy ve zlepšování nabídky ERP v orientaci na malé a střední podniky

Dosud byly systémy ERP používány především ve větších podnicích a organizacích. Ty jsou ale těmito produkty takřka nasyceny. Dříve bylo možné dobře identifikovat řešení ERP vhodná pro větší podniky a vedle nich i řešení pro střední a malé podniky. V současné době toto dělení postupně přestává platit a oblast malých a středních podniků se stává cílem všech dodavatelů ERP. Řešení pro malé a střední podniky však není jen jakýmsi zmenšením funkčních možností, ale vyznačuje se i osobitými metodami implementace a charakteristickým přístupem konzultantů dodavatelů. Mezi specifické požadavky menších podniků patří nižší cena, kratší doba implementace a větší tlak na přínosy řešení [24].

Systémy ERP představují jednu ze základních aplikací informační a komunikační techniky v podniku a zároveň i službu poskytovanou podnikovou informatikou uživatelům. Právě zajištění informačních potřeb uživatele v rámci podnikového procesu

je klíčové. To směřuje jednak ke snižování nákladů a současně ke zvyšování výnosů z realizovaných prodejů produktů a služeb [24].

Průzkumy ukazují, že se nabídka ERP na českém trhu velmi stabilizovala. Dá se říci, že se u nás etablovaly hlavní světové produkty ERP. Na druhé straně však stále existují produkty ERP, které v některých dalších evropských zemích dominují a pro nás jsou dosud takřka neznámé. Navíc se zlepšuje podpora rozhodování a rozšiřuje aplikační oblast do menších podniků [24].

Řešení ERP dnes neznamenaí jen společnou databázi údajů dostupnou všem uživatelům odkudkoli a kdykoli, ale představují nástroj pro zvyšování celkové efektivnosti fungování podniku, růstu jeho hodnoty a jeho finančních příjmů. Nastává doba, kdy velká část podniků bude muset v horizontu dvou až tří let uvažovat o inovaci stávajícího řešení ERP. Inovace by měla být příležitostí k uplatnění nového přístupu k ERP. To znamená nepovažovat ji pouze za nutnou investici, ale na základě vhodných kritérií daleko pečlivěji zvažovat její efektivnost a úroveň. Protože opět půjde o zásadní rozhodnutí s dlouhodobým efektem, bude důležité vycházet z nových trendů a nejen z přístupů a konceptů osvědčených v minulosti [24].

4.1.4 Možné alternativy řešení a jejich zhodnocení

K možným alternativám patří zakoupení a implementace již hotového IS stojí a padá s výběrem vhodného systému. Komplexní přehled ERP systémů je k nalezení na www.systemonline.cz [26]. Žádný systém samozřejmě nemůže být vybrán bez oslovení dodavatele k podání konkrétní nabídky (zejména posouzení, zda ERP systém má skutečně požadovanou funkcionalitu), proto je následující přehled jen podkladem pro budoucí kroky vedoucí k oslovení jednotlivých dodavatelů.

Jako poznámku je nutné uvést, že u těchto produktů není příliš ostře rozlišen nákup a pronájem: společnost většinou nemůže zakoupit produkt jako takový k neomezenému použití, nicméně platí za instalaci na určitý počet počítačů (a serverů), většinou po dobu jednoho roku.

ABRA G4, dodavatel ABRA Software a.s.

ABRA Software a.s. je výrobce a dodavatel nejrozšířenějšího ERP software v modelu klient/server v České republice. Firma má k dispozici více než sto zaměstnanců a

technické zázemí velké softwarové společnosti s ročním obratem přes 140 milionů Kč. Dodavatel je schopen pružně reagovat změnou zdrojového kódu software na změny legislativy či podnikových procesů. Dle přehledu na www.systemonline.cz [26] a www.abra.eu [27] byl vytipován software ABRA G4 – jako řešení pro střední a velké firmy, které se od softwaru G3 liší jen použitou databází Oracle, se kterou má IT oddělení firmy NOVASERVIS zkušenosti.

ABRA je univerzální informační systém pro řízení podnikových procesů. Nabízí využití více než 30 modulů zejména pro oblasti nákupu, výroby, prodeje, řízení vztahu se zákazníky, logistiky, účetnictví a financí, zpracování mezd a personalistiky. Počet současně pracujících uživatelů v systému se může pohybovat od jednotek až po stovky dle potřeby. ABRA využívá třívrstvou technologii klient/server se všemi výhodami, které přináší.

Cenu a dobu trvání implementace na svých stránkách dodavatel neuvádí, bude proto použit odhad lineární interpolací z předchozích zkušeností, který se samozřejmě může od skutečného stavu věcí více či méně lišit.

Náklady na pořízení software ABRA G4 pro 70 uživatelů

Licence	1,5 milionů Kč
Implementace	1 milion Kč
Roční náklady – 15% z hodnoty SW	225 000 Kč
Náklady první rok – cca 30 konzultačních dní	350 000 Kč
Celkem náklady na první rok	2,85 milionů Kč

Dodavatel při implementaci postupuje pomocí vlastní implementační technologie S.A.F.E., předpokládaná doba implementace je 15 – 20 měsíců.

Bílý motýl, dodavatel BM Servis spol. s.r.o.

Dle [28] BM Servis již 15 let patří na softwarovém trhu k nejstarším českým firmám, které se orientují na vývoj komplexních firemních informačních systémů pro společnosti podnikající v různých oblastech bez hranic a bariery jazyka. Společnost má několik desítek zaměstnanců, ročně dosahuje zisku zhruba 1,5 milionů Kč.

IS Bílý motýl je moderní integrovaný IS na podporu procesního a manažerského řízení ve středních firmách, nezávisle na druhu podnikatelské oblasti. Podporuje (informačně a znalostně) manažerské řízení a účetnictví firmy, umožňuje definici zavedených pracovních procesů, postihuje a řídí veškeré firemní aktivity, harmonizuje vnitropodnikovou komunikaci pracovních týmů a komunikaci firmy s okolím s využitím moderních forem komunikace. Pro manažerské řízení ve strategické a taktické rovině je informační podpora orientována na kvalitu analýzy již dříve získaných informací s důrazem na kvalitu prezentace a variantnost pohledů.

Cena produktu vychází z analýzy, bude proto opět odhadnuta.

Náklady na pořízení softwaru Bílý motýl pro 70 uživatelů

Licence (včetně db Sybase)	1,5 milionů Kč
Implementace	1,5 milionů Kč
Roční náklady – 15% z hodnoty SW	225 000 Kč
Náklady první rok – cca 30 konzult. dní	350 000 Kč
Celkem náklady na první rok	3,35 milionů Kč

Doba implementace je odhadnuta na 6 měsíců.

Informační systém K2, dodavatel K2 atmitec spol. s r.o.

Společnost K2 atmitec s.r.o., výrobce Informačního systému K2, vznikla v roce 1991. Po šestnácti letech působení na trhu informačních technologií patří mezi přední výrobce informačních systémů. Za dobu svého působení na trhu si společnost vybudovala širokou a z odborného hlediska vysoce funkční partnerskou síť, která zabezpečuje svým zákazníkům komplexní služby a servis. Partnerská síť je dlouhodobě a cíleně budována tak, aby byla zákazníkům co nejbližší a mohla na jejich požadavky a potřeby reagovat co nejoperativněji. Společnost zaměstnává špičkové programátory, vysoce kvalitní projektové manažery a konzultanty, kteří profesionálně řídí implementaci IS K2 do prostředí zákazníka. Společnost má zhruba 180 zaměstnanců, roční obrat téměř 350 milionů Kč [29].

Produkt K2 Enterprise je určen firmám, které chtějí mít procesní a organizační strukturu podporovanou informačním systémem. Tento produkt si pořizují společnosti, které od systému očekávají především automatizaci, zjednodušení a maximální

přizpůsobení svým podmínkám a specifikům. Efektivita spočívá v nahrazení uživatelských zásahů všude tam, kde je to možné. Zákazníci K2 Enterprise chápou jako prostředí, které jim umožní realizovat jejich představy a přitom zabezpečí požadované výstupy (ekonomika, sklad, řízení výroby). Ve většině případů jsou to firmy s vysokou úrovní logistiky a s velkým množstvím opakovaných činností.

Výhodou produktu K2 Enterprise je možnost použít libovolnou databázi – tedy i již firmou NOVASERVIS vlastněný Oracle a servisní střediska po celé ČR.

Náklady na pořízení softwaru K2 Enterprise pro 70 uživatelů

Licence	2,5 milionů Kč
Implementace	1,5 milionů Kč
Roční náklady – 15% z hodnoty SW	375 000 Kč
Náklady první rok – cca 30 konzultačních dní	350 000 Kč
Celkem náklady na první rok	4,35 milionů Kč

Doba implementace je odhadnuta na 10-12 měsíců.

KARAT Enterprise, dodavatel KARAT Software a.s.

Společnost KARAT Software, a. s., je významným výrobcem a dodavatelem komplexních informačních systémů a doprovodných služeb s výhradně českým kapitálem a více než patnáctiletými zkušenostmi ve svém oboru. Svými výkony, počtem pracovníků a spolupracujících partnerů, vybaveností a dalšími ukazateli se KARAT Software, a. s., řadí mezi přední české dodavatele informačních technologií. Společnost má opět více než 100 zaměstnanců, roční obrát společnosti je kolem 10 milionů Kč.

KARAT je komplexní podnikový informační systém, který využívá technologii Client/Server. Je určen zejména pro řízení středních a velkých výrobních a obchodních společností a také organizací podnikajících v sektoru služeb, které mají v úmyslu skloubit účetní, ekonomický, skladový a obchodní software do komplexního řešení na míru. Předností IS KARAT je podpora více platforem (MS, LINUX, UN*X a MS SQL, SYBASE, aj.) [30].

Pro cenu a dobu trvání implementace platí to samé, co pro systém ABRA G4

Náklady na pořízení softwaru KARAT Enterprise pro 70 uživatelů

Licence (včetně db Sybase)	5 milionů Kč
Implementace	2 miliony Kč
Roční náklady – 15% z hodnoty SW	750 000 Kč
Náklady první rok – cca 30 konzultačních dní	350 000 Kč
Celkem náklady na první rok	7,35 milionů Kč

Doba implementace je odhadnuta na 6 měsíců.

QI, dodavatel DC Concept a.s.

OR-NEXT spol. s r.o. je česká firma tvořena špičkovými odborníky, kteří ji díky svým nápadům a dlouholetým zkušenostem dostali až do stavu mezinárodně uznávaného tvůrce a dodavatele informačních systémů. V současné době díky tomu i díky široké partnerské síti systém používá v České a Slovenské republice již přes 520 firem a organizací. Dodavatel je díky velké elasticitě softwaru schopný pružně reagovat na změny. QI je elastický podnikový informační systém, který je ojedinělý svou celkovou koncepcí, vysokou koncentrací špičkových technologií a progresivní licenční politikou. Jeho základní vlastností je schopnost přizpůsobovat se změnám okolí a potřebám zákazníka za plného provozu. Navíc obsahuje vývojové prostředí pro rychlý vývoj a implementaci aplikací. QI není oborově orientován a je s úspěchem využíván subjekty všech velikostí, a to v oblastech obchodu, služeb, ekonomiky, výroby i státní správy. Výhodou pro společnost Novaservis je sídlo společnosti v Brně a tedy fyzická blízkost technické podpory [31].

Náklady na pořízení softwaru QI pro 70 uživatelů

Licence	3,25 milionů Kč
Implementace	3 miliony Kč
Roční náklady – 15% z hodnoty SW	500 000 Kč
Náklady první rok – cca 30 konzultačních dní	350 000 Kč
Celkem náklady na první rok	6,6 milionů Kč

Doba implementace je odhadnuta na 5 měsíců.

Závěrem těchto alternativ je to, že na trhu není cenově přijatelný informační systém. Mnohdy se stává, že firmy platí za něco, co doopravdy nepotřebují.

4.1.5 HOS analýza vhodnosti nasazení IS

Metoda HOS analýzy je založena na hodnocení tří komponent informačního systému - Hardware, Orgware a Software. Správné ohodnocení jednotlivých položek je důležité pro správnou klasifikaci informačního systému, ale na druhou stranu je toto hodnocení často velmi obtížné a je třeba se spokojit mnohdy jen s kvalifikovaným odhadem (obdobně jako u SWOT analýzy). Takže k zhodnocení vhodnosti IS/IT ve firmě Unicode systems:

- Hardware – výpočetní a komunikační technika je dostatečně výkonná pro provoz informačního systému. Jedná se o dobře vybavené počítače a dobře strukturovanou počítačovou síť. Firma disponuje i dostatečně vybavenými servery potřebnými pro provoz informačního systému.
- Software – co se týká programové vybavení, tak je plně taktéž dostačující. Potřeba jsou zejména počítače se systémem Windows a prohlížečem typu Internet Explorer a to je splněno.
- Orgware – vhodná a správná organizace IS/IT je neméně důležitá a je taktéž zajištěna

Z této analýzy vyplývá, že vhodnost nasazení informačního systému do firmy je příhodná.

4.1.6 Závěr analýzy nasazení informačního systému do firmy

Z analýzy současného stavu nasazení informačního systému do firmy vyplývá následující:

- nutnost nasazení informačního systému pro správu podniku je vhodná a příležitosti přesahují (např. snížení nákladů, zvýšení zisku) nad hrozbami
- na trhu není cenově přijatelný systém pro firmu Unicode systems
- vhodnost nasazení informačního systému do firmy je příhodná.

Takto se dostáváme k samotnému vlastnímu návrhu řešení, které je uvedeno v následující kapitole.

5 Vlastní návrh řešení

Celá tato kapitola je věnována návrhem informačního systému pro správu podniku a následném zhodnocení tohoto návrhu.

5.1 Analýza požadavků

Analýza požadavků je jednou z nejdůležitějších součástí procesu vývoje softwaru. Přesně specifikované požadavky nám pomáhají optimalizovat náklady (časové i finanční) na vývoj softwaru a jsou dobrým předpokladem vytvoření správnosti softwaru. Analýza požadavků analyzuje a definuje požadavky na software. Na základě analýzy je možné sestavit seznam požadavků, které musí informační systém splňovat a které uvádím v další podkapitole.

5.1.1 Požadavky na informační systém

- Přihlášení zaměstnance
- Prohlížení informací vlastního účtu
- Prohlížení informací o ostatních zaměstnancích
- Odesílání zpráv na Helpdesk (podpora problémů)
- Prohlížení stavu projektu
- Vyplňování reportů
- Zobrazení úkolů
- Měnění informací o zaměstnancích
- Měnění stavu projektu
- Rozdělování úkolů projektu
- Zadáávání projektů
- Přijímání zpráv na Helpdesk
- Odpovídání zpráv na Helpdesk
- Spravování databáze
- Registrace zaměstnance
- Uchování historie projektů
- Uchování historie zpráv helpdesku
- Neumožnění měnit informace nadřízeného

- Smazání zaměstnance ze systému

5.1.2 Role uživatelů

Navrhovaný informační systém rozlišuje čtyři role uživatelů. Nejvýše postavený je IT manažer (mohlo by se říct i třeba správce systému). Ten má nejvíce oprávnění. Dále může být v systému libovolný počet manažerů jednotlivých oddělení firmy. Pod manažery jsou vedoucí, kteří podléhají manažerům. A nakonec, nejnižší postavený je zaměstnanec. Jednotlivé možnosti a práva každého z nich uvedu níže. Samozřejmostí je, že jak IT manažer, tak i ostatní manažeři a vedoucí jednotlivých oddělení jsou zároveň zaměstnanci.

1. Zaměstnanec

- Každý zaměstnanec se může přihlásit do systému
- Dále může prohlížet informace o sobě
- Každý zaměstnanec může prohledávat seznam jeho kolegů. Vyhledávat jejich kontaktní údaje kromě soukromých informací
- Každý zaměstnanec může zasláním zprávy na helpdesk požádat o podporu při nějakém problému
- Taktéž může prohlížet stavy (průběh) projektů a zobrazovat úkoly, které mu byly přiřazeny
- Poslední možností je vyplňování reportů a jejich odeslání

2. Vedoucí:

- Každý vedoucí může prohlížet všechny informace o svých podřízených a může tyto údaje měnit
- Taktéž může měnit stav (průběh) zadaného projektu
- Navíc rozděluje a posílá úkoly svým podřízeným, co se týká projektu

3. manažer

- Všichni manažeři můžou zadávat projekty
- Můžou měnit všechny informace o svých vedoucích

4. IT manažer (správce systému)

- IT manažer spravuje databázi
- Přijímá zprávy na helpdesk od zaměstnanců
- Následně může odpovídat na zprávy na helpdesk
- Dále registruje uživatele a dělá celkovou správu uživatelů jako např. smazání uživatele apod.

5.1.3 Požadavky na uživatelské rozhraní

Uživatelské rozhraní (prostředí) v informačním systému by mělo být jednoduché, přehledné a intuitivní. Součástí systému by měla být jasná a srozumitelná nápověda, protože systém budou používat i obyčejní uživatelé a nejen počítačový experti. Proto musí být systém navržen tak, aby jej mohl hned intuitivně uživatel využívat a nebylo potřeba většího zaškolení. S kvalitním uživatelským prostředím se totiž snižují i náklady na školení zaměstnanců pro práci s informačním systémem.

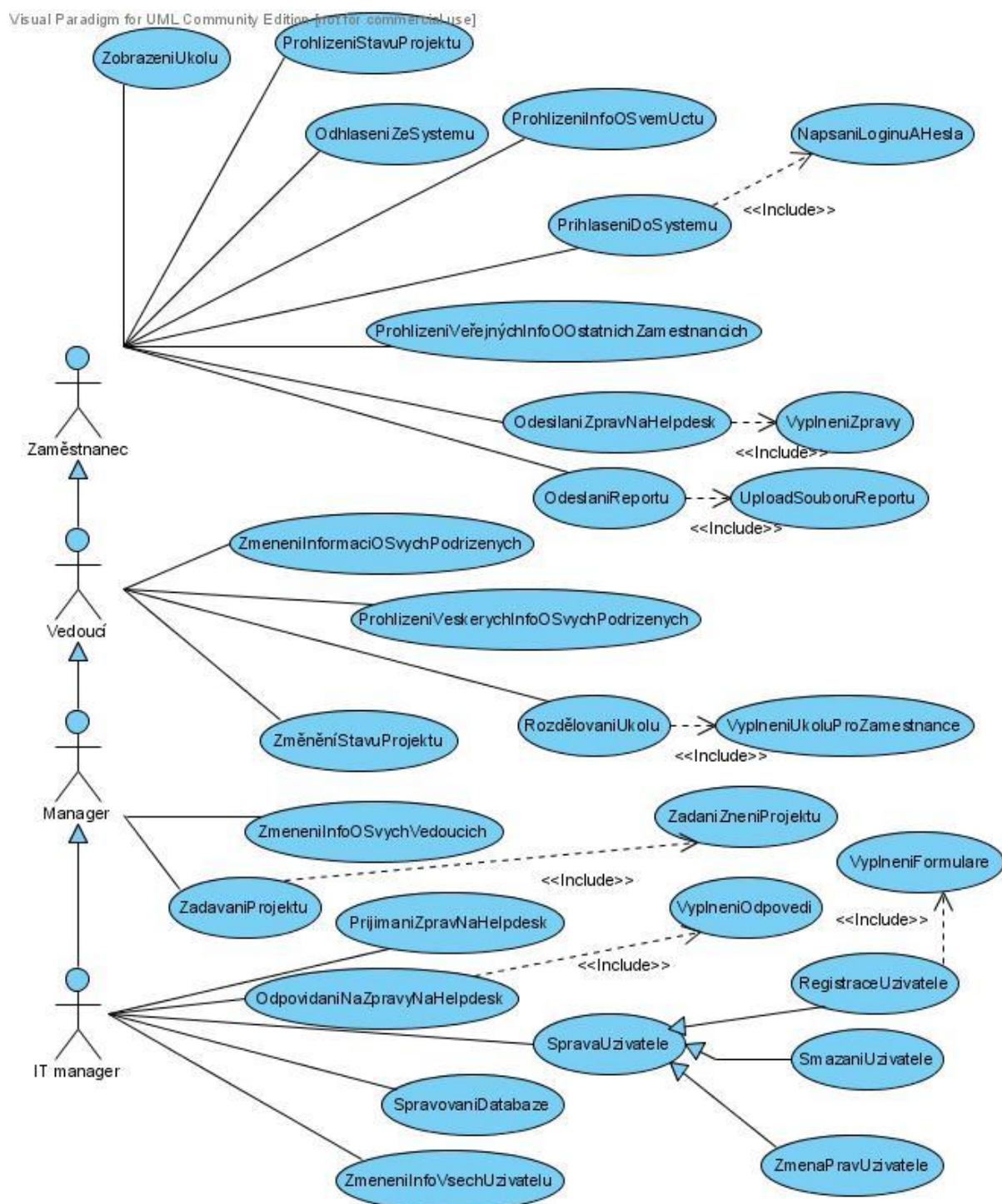
5.1.4 Use Case diagram

S odkazem na [1]. Use case diagram je zobrazení dynamické (funkční) struktury systému z pohledu uživatele. Je primárně určen k definici chování systému bez toho, že by odhaloval jeho vnitřní strukturu. Je to soubor scénářů pro používání systému. Každý scénář obsahuje:

- sekvenci (posloupnost) událostí, které v jeho rámci probíhají (včetně případných variant)
- popis interakce (komunikace) mezi uživatelem (aktorem) a systémem

Využití:

- specifikace požadavků na systém (podklad pro analýzu a návrh)
- komunikace se zákazníkem (uživatelem systému)
- podklad pro řízení projektu
- tvorba testovacích případů pro fázi testování systému (před uvedením do provozu)
- návrh uživatelského rozhraní (mezi každým use case a aktorem má být rozhraní)



Obrázek č. 5.1 – Use Case diagram informačního systému pro správu počítačového centra.

5.2 Návrh informačního systému

Původní vývoj softwaru začal prostým programováním. Jak rostla velikost systému a zkušenosti programátorů, tak si lidé uvědomili, že prostý zápis kódu aplikace nedostačuje. I kdyby taková aplikace fungovala, její kód by byl hůře udržitelný a rozšiřitelný. Seznamme se proto nejprve s návrhem.

Návrh softwaru umožnil vytvořit plán racionálního uspořádání kódu ještě před vytvořením prvního řádku do kódovací desky. Tato radikální změna dokonce umožnila lidem odhadnout potenciální problémy se správou už ve fázi přípravy podkladů.

Aby byly naplněny požadavky uživatele, byla také zavedena uspořádanější a přesnější analýza. Po celou historii softwaru se lidé snažili dosáhnout možnosti opakovaného používání kódu. Bohužel však většina procedurálních jednotek kódu není natolik uzavřena sama do sebe, aby je bylo možné nezávisle opakovaně používat. Avšak prostřednictvím objektové orientace máme další možnost dosáhnout opakovaného využívání. Historie objektové orientace je podobná hlavnímu softwarovému proudu. Práce s objekty však od implementace k abstrakci dosáhla nebývalého tempa. Objektově orientované programování nabylo na popularitě poprvé v 80. letech. Totéž desetiletí zažilo uvedení jak objektově orientovaného návrhu, tak i objektově orientované analýzy [2].

V této části se budu zabývat tedy objektově orientovaným návrhem a následně i objektově orientovanou implementací.

5.2.1. Objektová orientace

Než se podíváme na samostatný návrh v UML, tak popíšu nejdříve stavební kameny objektové orientace s odkazem na [2].

Objekty – spojují data a funkcionalitu společně do jednotek zvaných objekty, ze kterých se potom skládá výsledný objektově orientovaný program (na rozdíl od strukturovaného složeného z procedur a funkcí). Objekty jsou tedy základní jednotkou modularity i struktury v objektově orientovaném programu, který umožňuje problém intuitivně rozdělit na přímo realitě korespondující pododdíly a díky jejich vzájemné komunikaci i tento problém řešit.

Abstrakce – je schopnost programu ignorovat některé aspekty informací, se kterými pracuje. Abstrakce je pohled na vybraný problém reálného světa a jeho počítačové řešení. Při vytváření takovéto abstrakce je vhodné mít možnost skrývat detaily do jakési černé skříňky, která je pro okolí definovaná pouze svým rozhraním, přes které komunikuje s okolím, a nikoli vnitřními detaily implementace, která může být podstatným zjednodušením reálného světa.

Zapouzdření – zajišťuje již na úrovni definice sémantiky jazyka, tak že uživatel nemůže měnit interní stav objektů libovolným (tedy i neočekávaným) způsobem, ale musí k tomu využívat poskytované rozhraní (operace nad objektem). Každý objekt tedy nabízí protokol, který určuje, jak s ním mohou ostatní objekty komunikovat. Ostatní objekty se tedy při komunikaci s tímto objektem spoléhají pouze na jeho externí (poskytované) rozhraní a skutečná implementace zůstává dokonale skryta. Právě tento koncept zásadně zjednodušuje vývoj nových vlastností a vylepšování stávající implementace našeho programu, protože nám stačí zachovat pouze zpětnou kompatibilitu rozhraní objektů a nikoli skrytých implementací.

Polymorfismus – je mnohotvárnost využívající mechanismus zaslání zpráv. Namísto běžného volání podprogramů (procedur a funkcí) ve strukturovaném programování se v objektově orientovaném jazyce využívá konceptu zasílání zpráv. Konkrétní použitá metoda reagující na zaslání zprávy závisí na konkrétním objektu, jemuž je tato zpráva zaslána. Například, pokud máme objekt *orel* a pošleme mu zprávu *rychle_se_přemísti*, tak implementace reagující metody bude pravděpodobně obsahovat příkazy pro roztážení křídel a vzletnutí. Pokud ale budeme mít objekt *gepard*, tak implementace metody volané při obdržení tytéž zprávy bude obsahovat například příkaz pro rozběhnutí se. Obě reakce na zaslání stejné zprávy byly uzpůsobeny potřebám konkrétního objektu, který zprávu obdržel, a tudíž byly plně v jeho vlastní režii. Takto získáme polymorfismus, a tak může zaslání stejné zprávy vyvolat rozdílné reakce během různých kontextů.

Dědičnost – je způsob, jak implementovat sdílené chování. Nové objekty tak mohou sdílet a rozšiřovat chování těch již existujících bez nutnosti všechno znovu reimplementovat. Dědičnost se v praxi využívá ke dvěma účelům: k indikaci, že nové chování specializuje jiné již existující chování a pro sdílení kódu. Tato vlastnost je tedy velmi důležitá pro udržitelnost a rozšiřitelnost objektově orientovaných systémů.

5.2.2 UML

Veškeré návrhy a analýza budou prováděny pomocí prostředků UML, který se stává standardem v oblasti objektového návrhu.

Na konci 90. let Grady Booch, Ivar Jacobson a Jim Rumbaugh přišli s definitivní verzí unifikovaného modelovacího jazyka, který je nyní standardem přijatých kromě jiných také skupinou OMG (Object-Management Group). Zápis UML se podobá zápisu techniky modelování objektu (Object Modeling Technique – OMT), kterou Rumbaugh a další vyvinuli na začátku 90. let [2].

UML (Unified Modeling Language) je grafický jazyk pro vizualizaci, specifikaci, tvorbu a dokumentaci prvků softwarových systémů. UML je však využitelný i pro business modelování a pro modelování ne-SW systémů. V UML lze modelovat jakýkoliv typ aplikace běžící na jakémkoliv typu HW. Lze také modelovat SW nezávisle na operačním systému a programovacím jazyku. Pro UML je nejpřirozenější modelování pro objektově orientované prostředí, ale lze modelovat v jakémkoliv jiném jazyku. [2]

V současné době má jazyk UML největší význam při návrhu softwarových systémů, protože objektově orientovaný návrh každé složitější aplikace je nezbytným předpokladem pro její úspěšnou a rychlou implementaci. [2]

Pro objektově orientovaný návrh je samozřejmě možné použít různé podpůrné prostředky (zejména další odlišné typy diagramů). UML je však významné také v tom ohledu, že přesně specifikuje, co má daný diagram obsahovat, což je velmi důležité zejména při sdílení informací mezi jednotlivými analytiky a vývojáři. Dále je již z principu UML nutné, aby vytvářené grafy měly vnitřní konzistenci a přesně danou sémantiku. UML diagramů existuje několik typů lišících podle toho, jaké se pomocí nich plánují či zpracovávají úlohy. Tyto diagramy se od sebe odlišují především repertoárem použitých značek, způsobem jejich vzájemného propojení a s nimi související sémantikou. [2]

5.2.3 Softwarová architektura

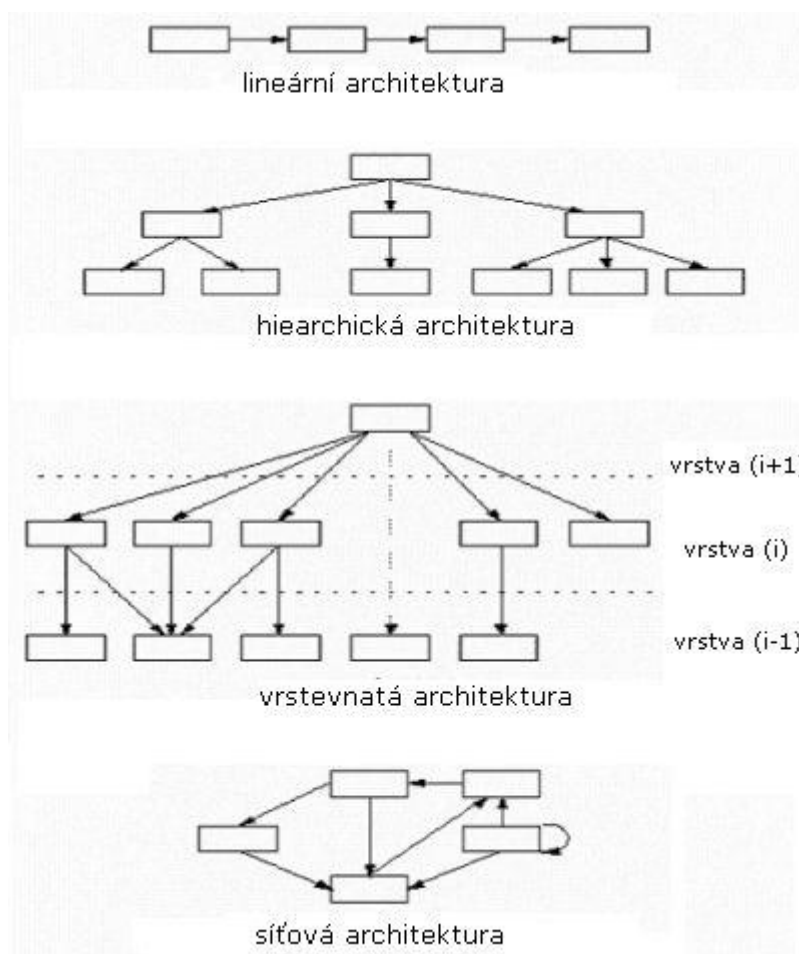
S rostoucí velikostí a složitostí softwarových systémů se návrh a specifikace celkové struktury systému stává významnějším problémem než volba algoritmů a datových

struktur. Softwarová architektura je struktura komponent programu/systému, jejich vzájemné vazby, principy a předpisy určující jejich návrh a vývoj v průběhu času.

Z pohledu našeho zájmu lze činnosti prováděné při návrhu programového systému rozdělit do několika skupin. Jednu skupinu tvoří činnosti, které provádějí dekompozici systému do dekompozičních jednotek (míněno z pohledu návrhu) v potřebném počtu úrovní dekompozice. Činnosti, které provádějí specifikaci těchto dekompozičních jednotek. Činnosti navrhující spolupráci těchto dekompozičních jednotek. Tedy činnosti jednoznačně zřejmé, když se hovoří o návrhu. Pracovně tyto činnosti nazvěme vlastní návrh. Vlastní návrh má jednoznačnou návaznost na analýzu vyvíjeného systému v tom smyslu, že navrhuje realizaci analytického modelu vyvíjeného systému. Architektura programového systému má zásadní vliv jak na jeho fungování, tak na jeho vývoj, údržbu a další rozvoj. Uvádím zde stručný rozbor přínosů a možností, které softwarová architektura nabízí:

- dokumentace softwarové architektury představující jednotný přístup k strukturálním otázkám je nesmírně přínosná pro celý další život systému, tj. jeho vývoj a (hlavně) údržbu.
- softwarová architektura by měla systematicky odrážet kvalitativní nároky na vyvíjený systém (např. udržovatelnost, rozšiřitelnost) a naopak analýzou softwarové architektury lze o systému mnohé říci, tedy jde o jednoznačný přínos k zajišťování jakosti softwarového procesu a produktu.
- vědomé stanovení softwarové architektury nutí k jednotnému a vnitřně konzistentnímu učinění těch nejzásadnějších rozhodnutí designu.
- stanovená a dokumentovaná softwarová architektura umožní její sdělení všem zainteresovaným stranám při vývoji a údržbě softwarového systému.

Obrázek č. 5.2 podle zdroje [21] znázorňuje typy softwarových architektur.



Obrázek č. 5.2 – Typy architektur

5.2.4 Návrhové vzory

Základním kamenem pro návrhové vzory v oblasti informační technologie se stalo dílo *Design Patterns: Elements of Reusable Object-Oriented Software*. Autorem bylo seskupení zvané Gang of Four (pánové Gamma, Helm, Johnson a Vlissides) a poprvé bylo představeno na OPSLA'94, kde se setkalo s velkým úspěchem. Tato práce je nazývána základní knihou v oblasti návrhových vzorů a dodnes se jedná o velmi uznávané dílo. Většina dnešních děl o návrhových vzorech vychází z rozdělení, které v tomto díle bylo popsáno. Vzhledem k tomu, že tato kniha si udržuje svoji hodnotu i v rychle se vyvíjejícím světě informačních technologií, jedná se opravdu o unikátní dílo. [3]

Návrhové vzory jsou doporučené postupy řešení často se vyskytujících úloh. Můžeme je považovat za vzory, které použijeme při návrhu architektury budoucí aplikace. Výrazně se sníží pravděpodobnost chyb, které uděláme, když bychom řešení sami

vymysleli. Současně si často ulehčíme budoucí práci, protože návrhové vzory již dopředu počítají s typickými rozšířeními. Takže budoucí úpravy a rozšíření dají mnohem méně práce a výsledné programy budou navíc spolehlivější.[4]

Vzory lze pochopitelně využít i v jiných fázích vývoje softwaru než je návrh. Existují například analytické vzory používané ve fázi analýzy. Tato práce využívá softwarovou architekturu Model – View – Controller (MVC). O této architektuře si řekneme více v další kapitole.

S odkazem na [20] zmíníme v přehledu návrhové vzory. Toto rozdělení vychází z výše zmíněné knihy Design Patterns: Elements of Reusable Object-Oriented Software. Autoři vymezili tři základní skupiny návrhových vzorů, které jsou stále uznávané. Návrhové vzory se rozdělují do tří skupin a to: Creational Patterns (vytvářející), Structural Patterns (strukturální) a Behavioral Patterns (vzory chování).

Creational Patterns řeší problémy související s vytvářením objektů v systému. Snahou těchto návrhových vzorů je popsat postup výběru třídy nového objektu a zajištění správného počtu těchto objektů. Většinou se jedná o dynamická rozhodnutí učiněná za běhu programu. Mezi tyto návrhové vzory patří: Factory Method Pattern, Abstract Factory Method Pattern, Builder Pattern, Prototype Pattern, Singleton Pattern.

Structural Patterns představují skupinu návrhových vzorů zaměřujících se na možnosti uspořádání jednotlivých tříd nebo komponent v systému. Snahou je zpřehlednit systém a využít možností strukturalizace kódu. Mezi tyto návrhové vzory patří: Adapter Pattern, Bridge Pattern, Composite Pattern, Decorator Pattern, Facade Pattern, Flyweight Pattern, Proxy Pattern.

Behavioral Patterns se zajímají o chování systému. Mohou být založeny na třídách nebo objektech. U tříd využívají při návrhu řešení především principu dědičnosti. V druhém přístupu je řešena spolupráce mezi objekty a skupinami objektů, která zajišťuje dosažení požadovaného výsledku. Mezi tento typ vzorů můžeme zařadit: Chain Of Responsibility Pattern, Command Pattern, Interpreter Pattern, Iterator Pattern, Mediator Pattern, Memento Pattern, Observer Pattern, State Pattern, Strategy Pattern, Template Pattern, Visitor Pattern.

5.2.5 Architektura MVC

Hlavním bodem této práce a návrhu je softwarová architektura Model – View – Controller (Model – Pohled – Řadič). Tato architektura je ve velké míře využívána při vytváření prezentačně orientovaných systémů a může být publikována jako návrhový vzor.

MVC odděluje část programu mající na starost před zpracováním příkazu uživatele od části zabezpečující logiku programu, která uživatelské příkazy zpracovává a části, která má na starosti zobrazení výsledků. MVC se stává ze tří druhů objektů: model představuje objekt aplikace, view (pohled) představuje prezentace na obrazovce a controller (ovládání) definuje způsob, jak rozhraní reaguje na vstup uživatele. MVC rozděluje objekty systému a tím zvyšuje flexibilitu a znovu-použitelnost celého řešení. Do části zabývající se prezentací výsledků patří nejenom vlastní GUI ale často i kód, který data na tuto prezentaci připravuje. Aplikace tohoto vzoru umožňuje snadnou implementaci zadávání požadavků různými způsoby (klávesnice, myš, hlas). Stejně jako různé možnosti prezentace výsledků (tabulka, graf, mluvené slovo). Další výhodou takového rozdělení je usnadnění případných budoucích změn. Aplikace vzoru usnadňuje přenositelnost mezi platformami.

Vzor je spíše obecným doporučením, které jednotlivé vazby nijak konkrétně nespecifikuje. V jednodušších aplikacích se některé části často slučují. Obdobně je to i při použití vzoru v jiných situacích. Např. pokud navrhujeme lepší algoritmy na zpracování dat, tak inovujeme model a zachováme-li komunikační rozhraní se zbylými dvěma částmi, uživatel se o tom vůbec nedozví. Dostali jsme požadavek zobrazovat data v různých formátech. Definujeme čtyři různé pohledy a s využitím vzoru nechme na uživateli, aby si vybral, který je pro něj v dané chvíli optimální. Oddělíme-li dobře jednotlivé části aplikace, můžeme získat aplikaci, která je velice snadno přenositelná mezi platformami. Aplikace mohou být naprogramovány na platformu JAVA SE a aplikace, která pro zobrazení používá knihovnu TVARY, lze prostou výměnou této knihovny převést na platformu JAVA ME, takže si je pak budeme moci pouštět třeba na mobilním telefonu. [4]

Tento návrhový vzor je takovým obecným doporučením, které je možné implementovat různými způsoby. Vzor například nijak nespecifikuje, jak přesně má komunikace mezi jeho jednotlivými částmi probíhat. V některých aplikacích posílá model v zobrazovacím modulu informace o tom, kdy a jak je má zobrazit. Jindy si

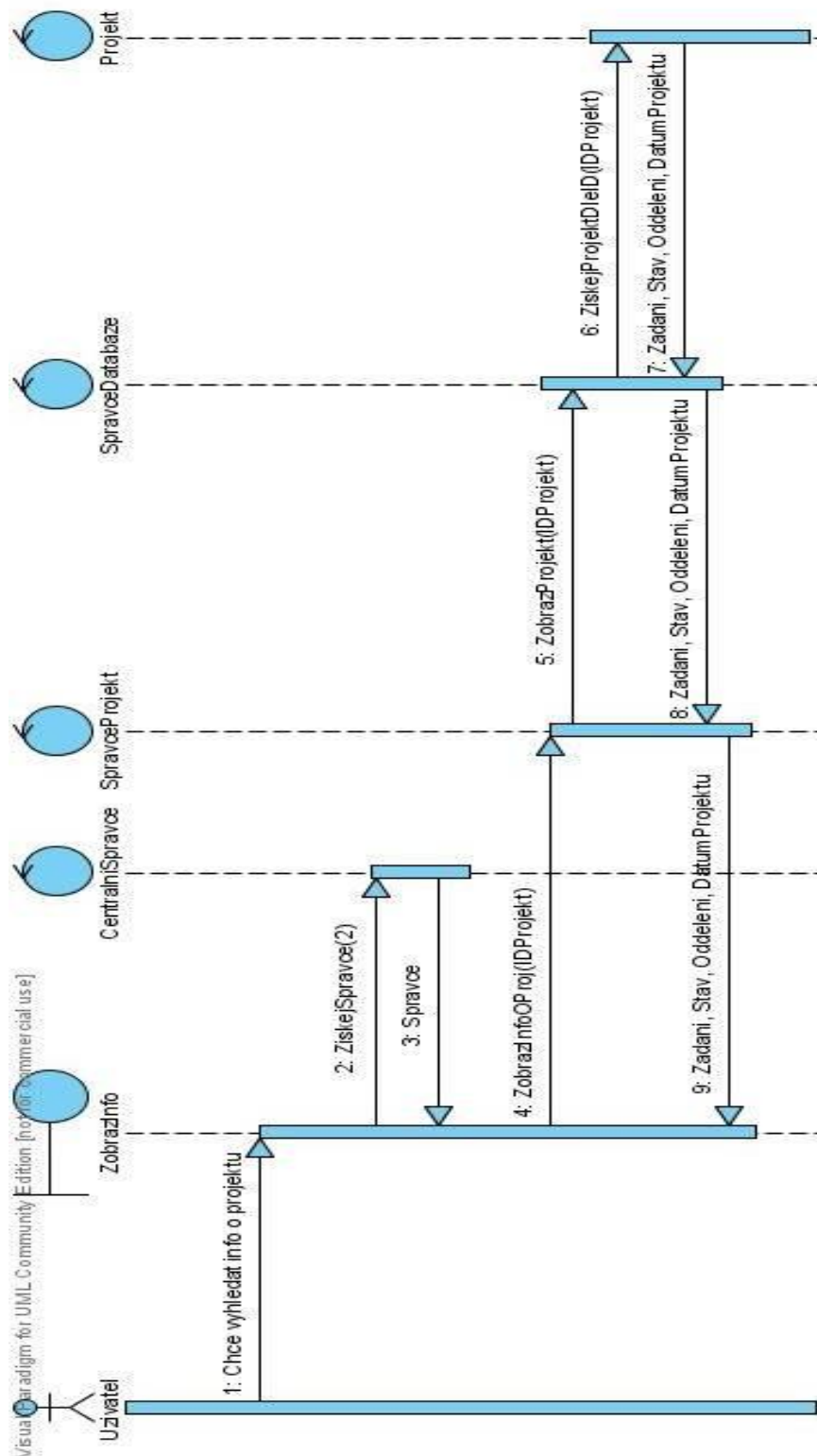
naopak zobrazovací modul o tyto informace sám říká modelu. Jak jsem již uváděl, v řadě jednodušších aplikací se někdy zobrazovací a ovládací část slučují do jedné třídy. Protože použité grafické uživatelské rozhraní k tomu přímo vybízí. Stručně řečeno: nelze najít žádná přesná implementační doporučení. Pouze obecné zásady doporučující tyto tři činnosti oddělit, abych pak mohl s minimální námahou změnit kteroukoli z nich, aniž bych tím ovlivnil zbylé dvě. Taktéž často se používá aplikace vzoru pozorovatel. Pohled se přihlásí u modelu jako jeho pozorovatel a model jej v případě jakékoliv změny na tuto skutečnost upozorní. Pohled si pak od modelu vyžádá potřebné informace a aktualizuje výstup [4].

Na další straně je obrázek č. 5.3, který ukazuje výsledné navržení systému v architektuře MVC. Začnu popisem části zvané Model. V této části je 6 tříd. A to Uživatel, který může odesílat Report, ten představuje další třídu. Poté Uživatel může zadávat Projekt a Úkol, což jsou další dvě třídy. V neposlední řadě Uživatel komunikuje s třídou OdpověďNaZprávuHelpdesku a DotazNaHelpdesk. Je také využito dědičnosti mezi třídami OdpověďNaZprávuHelpdesku a DotazNaHelpdesk. Každá z těchto tříd má své atributy. Všechny tyto třídy komunikují se třídami části zvané controller pomocí třídy SprávceDatabáze, což je třída controlleru. Třídami controlleru jsou SprávceÚkol, SprávceUživatel, SprávceProjekt, SprávceHelpdesk, SprávceUživatelAdmin, SprávceReport a každá z těchto tříd implementuje vlastní interface, který má stejné metody. Každá metoda je ve třídě vyjádřena názvem, který přibližuje její chování. Např. metoda *ZkontrolujLoginAHeslo* má na starost kontrolu hesla a loginu přihlašujícího uživatele. Pro spojení a vymezení tříd části zvané view a části controller je tu třída controlleru *CentrálníSprávce*. V části view jsem navrhnul 8 tříd. Např. třída Přihlášení je zobrazení přihlašovacího okna anebo třída ZobrazProjekt je zobrazení projektu podle ID na obrazovce.

5.2.6 Sekvenční diagram

Pro názornost a návrh časování informačního systému jsem si vybral jeden z diagramů jazyka UML a to diagram sekvenční. Je to typ diagramu interakce objektu, který zdůrazňuje časovou posloupnost vztahů mezi statickými objekty. S pomocí sekvenčního diagramu je možné jednoznačně určit, jaké zprávy si objekty zasílají. Kdykoli se stanou procedurální nebo časovací záležitosti nejasnými, je vhodné použít tento diagram pro vyjádření časové posloupnosti vztahů. Jak ukazuje sekvenční diagram, čas se zvyšuje směrem dolů po svislé ose a seznam objektů, které budou přijímat zprávy, se rozšiřuje podél vodorovné osy horní části diagramu. Tyto objekty mohou být uvedeny pod svými názvy tříd, je-li to dostatečně přesné. Tělo diagramu zachycuje aktivované operace s velikostí odrážející jejich přibližné trvání: symbol aktivované operace svisle přibližně proporcionálně odpovídá době, po kterou je operace aktivní. Šipky mezi operacemi popisují zprávy. Volitelně použitelná sekvenční čísla na zprávách vyznačují časovou posloupnost.[3]

Na následující straně je obrázek č. 5.4, který ukazuje sekvenční diagram, který demonstruje vyhledání informací zadaného projektu v rámci MVC architektury. Uživatel chce vyhledat informace o projektu. View ZobrazInfo nejdříve získá Správce metodou *ZiskejSprávce(2)*. Poté pomocí view ZobrazInfo a metody *ZobrazInfoOProj()* si vyžádá zobrazení informací o projektu podle zadaného ID přímo třídě controlleru SprávceProjekt. V dalším kroku SprávceProjekt zavolá funkci *ZobrazProjekt()*, která v další třídě controlleru SprávceDatabáze zavolá metodu *ZiskejProjektDleID()*. Tato funkce poté v třídě modelu zpět pošle ostatní atributy (Zadání, Stav, Oddělení, DatumProjektu) projektu dle dříve zadaného ID. Poté tyto atributy se navrací zpět až view ZobrazInfo. A uživateli se zobrazí hledané informace o projektu.



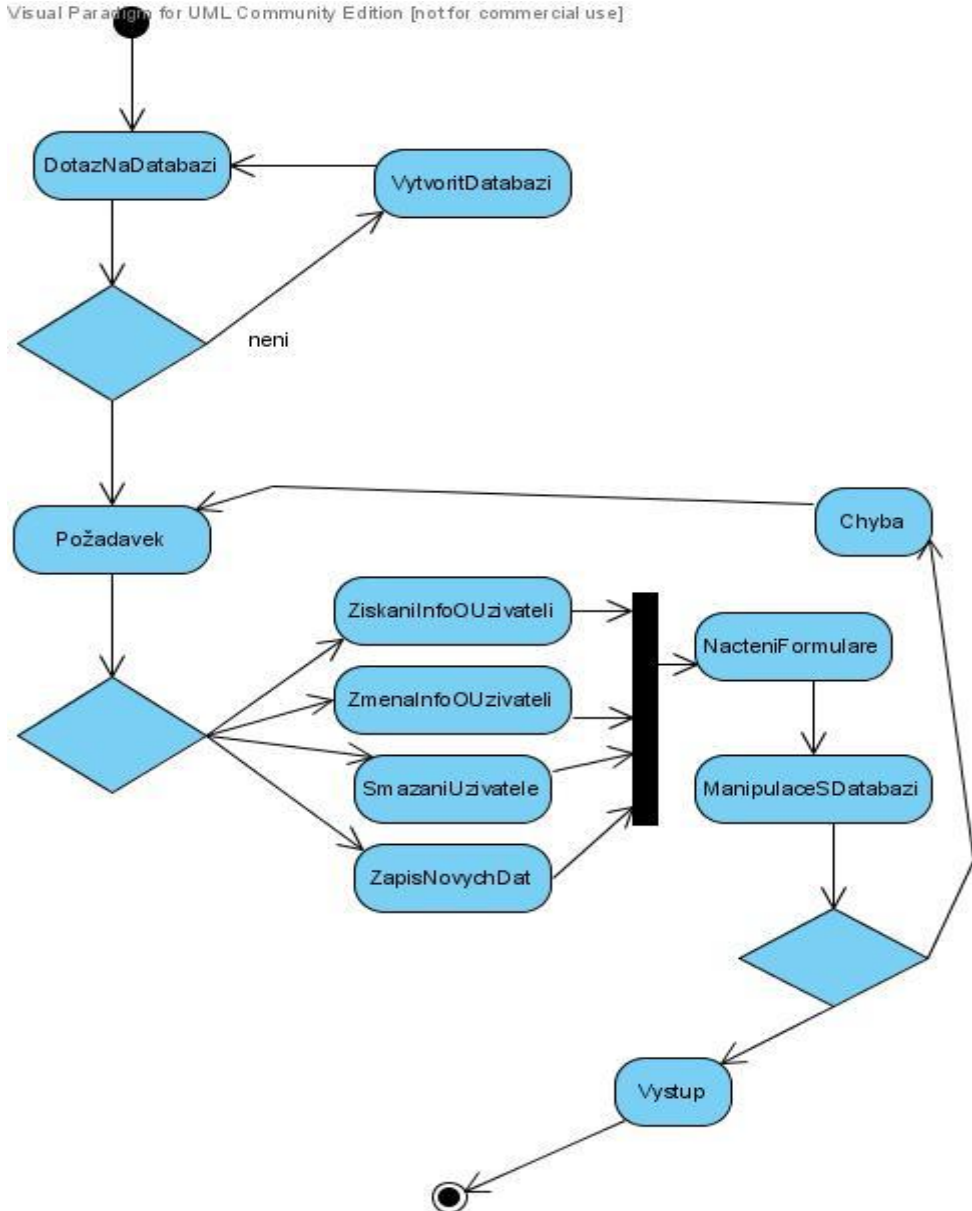
Obrázek č. 5.4 – Sekvenční diagram.

5.2.7 Diagram aktivit

S odkazem na [5]. Diagram aktivit je reprezentace struktury (dynamiky) počítačových a organizačních procesů v systému, zaměřená především na jeho vnitřní chování. Je to zobrazení řídicích toků (přechodů) mezi akcemi (aktivitami) v systému od počátečního bodu po jeden nebo více koncových bodů. Důraz se klade na zobrazení pořadí aktivit. Využití:

- modelování průběhu jednotlivých Use Case a operací v třídách
- modelování podnikových procesů (business process modeling, business modelování)
- workflow

Následující obrázek č. 5.5 demonstruje diagram aktivit části navrženého systému pro akce spojené s databází a tabulkou uživatel. Po vstupním bodu je akce *DotazNaDatabázi*. Poté je rozhodnutí o existenci databáze, a pokud databáze neexistuje, nastane akce *VytvořitDatabázi*. Pokud databáze existuje, tak se přikročí k *Požadavku* na databázi. Poté zde může nastat několik akcí – *ZískáníInfoOUživateli*, *ZměnaInfoOUživateli*, *SmazáníUživatele*, *ZápisNovýchDat*. Po některé z těchto akcí se *NačteFormulář* a zkontroluje se *ManipulaceSDatabází*. Pokud není manipulace v pořádku, tak nastane akce chyba a přejde se zpět k akci *Požadavek*. Pokud je manipulace v pořádku, tak následuje *Výstup*.



Obrázek č. 5.5 – Diagram aktivit.

5.3 Implementace

Nyní je proveden již návrh informačního systému, a proto se podíváme na možnosti vytváření dynamických webových stránek. Navrhovaný systém bude vyvíjený jako webová aplikace. Postupně vyjmenuji a popíši stručný přehled technologií, které jsem použil při implementaci. Rozlišujeme dva hlavní přístupy: technologie na straně klienta a technologie na straně serveru. S odkazem na [8] si uděláme přehled těchto technologií.

5.3.1 Technologie na straně klienta

Technologie na straně klienta jsou velice často inovovány. Pro implementaci webových stránek je to velická nevýhoda, neboť správce webového serveru nemůže ovlivnit software u každého klienta. Firmy chtějí obsáhnout největší množství uživatelů s různými prohlížeči. Tato skutečnost brání přijímání nových technologií, protože je podporují pouze nejnovější prohlížeče. Obecně se tyto technologie hodí pro nepříliš složité stránky, které nevyžadují speciální funkčnost.

Technologie na straně klienta přináší jednu výhodu. Jelikož jsou stránky zobrazovány (graficky interpretovány) na klientově systému, nezatěžují server. Jedná se např. o kontrolu formulářů, změny vzhledu stránek atd. Možné technologie jsou

- JavaScript
- XHTML
- Cascading Style Sheets (CSS)

5.3.2 Technologie na straně serveru

Technologie na straně serveru nabízí, oproti výše uvedeným způsobům, řadu výhod. Dochází k minimalizaci provozu na síti, díky omezení potřebné komunikace prohlížeče a serveru. Server zpracuje požadavek, vykoná potřebné úkony a odešle čistou XHTML stránku. To v konečné fázi znamená rychlejší načtení stránky a vyřešení problému s kompatibilitou prohlížečů. Dnes již by každý prohlížeč měl podporovat XHTML standardy. Další nespornou výhodou je možnost práce s daty na straně serveru. Server má k dispozici data a může je lehce poskytnout prohlížeči. Bezesporu největší výhodou je zabezpečení. Můžeme jednoduše zakódovat věci, které prohlížeč nikdy neuvidí.

Technologií na straně serveru existují více.

- Common Gateway Interface (CGI)
- API rozhraní webového serveru
- Active Server Pages (ASP)
- JavaScript na straně serveru
- Servlety a Java Server Pages (JSP)
- PHP: Hypertext Procesor (PHP)

V naší práci využijí pro definici dokumentu jazyk XHTML. Pro grafické upravení využijeme skriptovací jazyk JavaScript a kaskádové styly CSS. Funkčnost a práci s daty nám zajistí skriptovací jazyk PHP a databázový jazyk MySQL.

5.3.3 Použité prostředky

Dnes se už skoro žádná stránka neobejde jen s použitím klasického XHTML. Čas statických webových stránek je dávno pryč.

Systém správy pro počítačové centrum byl implementovaný jako webová aplikace hlavně pomocí dvou technologií a to PHP a MySQL.

Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a APACHE. Jednou z výhod použitých prostředků je v tom, že uživatel nepotřebuje žádný speciální software, aby mohl informační systém využívat. Postačí mu obyčejný webový prohlížeč. Taktéž údržba je mnohem jednodušší.

5.3.3.1 PHP

Začátky PHP se datují rokem 1994, kdy jeden nezávislý dodavatel vyvíjející software jménem Rasmus Lerdorf vyvinul jistý skript Perl/CGI, který mu umožňoval zjistit, kolik návštěvníků čte jeho online resumé. Aby spouštění perlu tolik nezatěžovalo server, přepsal ho do jazyka C. Tento systém se stal ihned populárním, a proto ho autor rozšířil a uvolnil pod názvem Personal Home Page Tools, poté Personal Home Page. Postupně narůstali užitečnosti a možnosti této technologie, došlo i ke změně slov udávajících význam PHP. V poslední verzi PHP a to verzi 5.0 se PHP přiblížilo ostatním jazykům podporující objektově orientované programování [9].

PHP je dnes velmi rozšířená technologie, umožňující jednoduché programování na straně serveru. Je to serverový skriptovací jazyk (server-side) navrhnutý pro potřeby webových stránek. To znamená, že všechno co PHP vykoná, neprobíhá na straně klienta, jako je tomu například u JavaScriptu, ale interpretuje se na straně serveru. Vygeneruje stránku podle zadaných kritérií a výsledek, který uvidí uživatel, odešle volajícímu počítači stejným způsobem, jakým se odesílají běžné statické stránky. To je možné využít k tvorbě různých interaktivních webových stránek. Takto je možné vytvořit různá vylepšení, od těch jednodušších jako jsou různé ankety nebo i knihy návštěv až k složitějším redakčním systémům, grafickým aplikacím či aplikacím s elektronickou poštou.

V PHP je podporováno propojení s velkým množstvím databází. Dalším podporovaným protokolem je LDAP v podobě API rozhraní klientských programů. PHP podporuje také jazyk XML, který je považován za jazyk budoucnosti. Dále podporuje protokoly IMAP a SMTP pro využívání mailových služeb serverů. Jazyk PHP zahrnuje podporu pro obrázky, PDF dokumenty a síťové služby. Výhodou jazyka PHP je, že lze vložit jednotlivé přídatné moduly, které mohou obsluhovat jakýkoliv protokol nebo práci s daty.

Z uvedeného vyplývají hlavní přednosti PHP – každý den můžeme aktualizovat obsah stránek na serveru tím, že změníme data v databázi a ne tím, že budeme každou stránku samostatně přepisovat.

Jazyk PHP se skládá se základních řídicích struktur, operátorů, druhů proměnných, deklarací funkcí, deklarací tříd, objektů, atd. Patří mezi jazyky netypované, dále jakákoliv proměnná zde může kdykoliv změnit svůj typ [9].

5.3.3.1.1 Výhody jazyka PHP

- výkon
- těsná integrace s většinou databázových systémů
- stabilita
- je zdarma
- nejsou problémy s kompatibilitou prohlížečů
- prohlížeč může poskytnout data, která nejsou na straně uživatele k dispozici
- open source – volně šiřitelná technologie
- není závislý na platformě a není vázaný žádným konkrétním serverem – může tedy běžet kdekoliv
- objektově orientovaný programovací jazyk, který je svojí strukturou podobný jazyku C++
- lepší zabezpečení oproti HTML

5.3.3.1.2. Nevýhody jazyka PHP

- jestliže je stránka načítaná, pomocí PHP ji už není možné dále měnit

5.3.3.2 SQL

Databázi je v podstatě možné si představit jako prostor, do kterého se ukládají všechny potřebné údaje. Zpracováním a přístupem k databázi je pověřený program, kterému se říká Systém řízení báze dat (SRBD) [17]. Představitelem SRBD je například MySQL. Většina dnešních SRBD je založena na tzv. relačním modelu dat. Název tohoto modelu vychází z relační algebry, což je matematický aparát, na kterém relační model staví. V tomto modelu jsou údaje uspořádané do tabulek [17].

Při práci s SRBD programem se používá model klient/server. Jako server vystupuje SRBD program poskytující svoje služby. Je tak nepřetržitě spuštěný a čeká na požadavky od klientů – jejich aplikací [17].

Pro zápis těchto požadavků na databázový server se dnes nejčastěji používá jazyk SQL. Je to zkratka znamenající Structured Query Language. Samotný SQL obsahuje všechno potřebné pro práci s databázemi – vytváření, rušení, modifikování tabulky, ale i nástroje pro práci se samotnými údaji – přidávání, změnu, rušení a vyhledávání údajů.

SQL se stal univerzálním jazykem pro programování databází. Pomocí tohoto jazyka určujeme, co chceme v databázi vykonat [17].

5.3.3.3 MySQL

MySQL je multiplatformní databáze. Komunikace s ní probíhá, jak už název napovídá, pomocí formy jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. Je to systém, který se etabloval především ve webových aplikacích, a který je hodně preferovaný při spolupráci s PHP.

Ve svojí podstatě je MySQL ořezaný o některé možnosti, které mají jiné databázové systémy. Důsledkem je nenáročnost MySQL na zdroje počítače a zvýšení rychlosti u některých operací. MySQL je jednoduše rychlý, jednoduchý a nenáročný na databázový systém. Níže uvádím další výhody.

5.3.3.3.1 Výhody MySQL

- open source
- cachování dotazů – uchovává si dotazy SELECT spolu s výsledky. Při následných dotazech jich porovnává s těmito cachovanými dotazy. Jsou-li shodné, MySQL

obejde nákladné získávání dat z databáze, a prostě použije výsledky cachovaného dotazu

- fulltextové indexování a vyhledávání
- replikace – umožňuje, aby se databáze umístěná na jednom serveru MySQL zduplikovala na jiný, což přináší množství výhod
- zabezpečovací a konfigurační volby

5.3.3.4 Webový server Apache

Webový server Apache je nejrozšířenější webový server, i když podle statistik se jeho využívání snižuje, tak je stále v provozu asi na 50 % serverech na internetu. Velkou výhodou je, že to je to open source projekt a je tedy zdarma. Mezi jeho přednosti patří možnost použití na mnoha operačních systémech – Linux, BSD, Microsoft Windows, je neustále vyvíjen a obsahuje nejnovější standardy. Apache je server velice dobře konfigurovatelný a může být doplněn o různé moduly i vlastní výroby.

5.3.3.5 XHTML a CSS

Dokument XHTML (Extensible HyperText Markup Language) je rozšiřitelný značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na internetu. Jeho historie začíná příchodem HTML a datuje se k roku 1991. Je složen ze základních příkazů, kterým se říká tagy. Píše se v ostrých závorkách, tedy mezi znaménky menší (<) a větší (>), a dělíme je do dvou skupin — na tagy párové a nepárové [18].

Každý tag má speciální význam. Ovlivňovat chování tagu můžeme pomocí parametrů a jejich hodnot. Tyto parametry jsou uvedeny za názvem tagu. Př. . Tento tag vypíše text uvedený mezi značky tagu o velikosti písma 2.

Jednotlivé tagy lze zanořovat. Platí ovšem pravidlo, že tagy se nesmí vzájemně křížit. Nesmíme tedy ukončit nadřazený tag, dříve než ukončíme tag vnořený.

CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích XHTML. CSS umožňují přiřadit současně mnoho vlastností všem prvkům na stránce, které jsou označeny konkrétním tagem. Hlavní výhodou je oddělení definice vzhledu od definice kostry stránek. Jsou tu ale také nevýhody. Největší nevýhodou je, že CSS nejsou plně podporovány žádným internetovým prohlížečem. CSS se skládají z

tagu neboli selektoru a definice neboli deklarace nebo více definicí. Definice je uzavřena ve složených závorkách { } a mezi jednotlivými definicemi se píše středník [18].

Kaskádové styly lze rozdělit do dvou skupin, na interní a externí. Interní se zapisují v sekci READ. Vložení je způsobeno zápisem tagu <STYLE></STYLE>. Mezi tento tag se poté vkládají jednotlivé definice. Zápis externích CSS se provádí pomocí tagu <LINK RE#=" stylesheet" TYPE="text/css" HREF=" soubor.css">. Samotné definice jsou uvedeny v externím souboru soubor.css a zapisují se stejně jako v interní definici.

5.3.3.6 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk. Ale na rozdíl od PHP se kód vykonává na straně klienta. Kód, který se má vykonat, se klientovi pošle uvnitř HTML stránky a kód vykoná prohlížeč během zpracovávání stránky.

JavaScript se používá pro generování dynamického obsahu podle různých okolností. Např. vložení aktuálního času, reakce na verzi prohlížeče. Také umožňuje interaktivní práci s dokumentem, reakce na události (pohyb myši, kliknutí, změna hodnoty) a v neposlední řadě může sloužit k ovládání prohlížeče (pohyb v historii, otevírání nových oken, ovládání stavového řádku). Výhoda spočívá v tom, že není nutné pro změnu stránky neustále obnovovat obsah stránky. Omezení však spočívá v možnostech použití, taktéž i v jejich (ne)bezpečnosti. Protože skripty se vykonávají přímo na počítači uživatele, mohl by potencionální útočník bez problémů vykonat nebezpečný kód. Proto jsou v JavaScriptu vypnuté funkce, které by přímo pracovaly se soubory na disku. JavaScript se kvůli tomuto omezení taktéž používá jako doplněk při vytváření webové grafiky. V této práci je například JavaScript použit jen k zobrazení aktuálního času. I když čas je získávaný z počítače klienta, výhoda je v jednoduchosti implementace na rozdíl od PHP, kde je tato operace složitější [19].

5.3.4 Implementace uživatelského rozhraní

Při vstupu do systému se musí každý uživatel nejprve přihlásit. Proto první stránka obsahuje pouze odkaz na přihlašovací formulář. Přihlašuje se zadáním uživatelského jména a hesla. Pokud uživatel nezadá platné údaje, tak mu není povolen vstup do systému a je vyzván k vložení nových údajů. Jedním ze způsobů, jak implementovat autentizaci uživatele, je použití standardní HTTP autentizace. Protokol HTTP nabízí

jednoduché prostředky pro ověření totožnosti uživatele. Pokud uživatel požádá server o nějaký prostředek, k němuž je přístup omezen, odešle server klientovi zprávu 401 (neautorizovaný přístup). Klient tuto zprávu rozpozná a zobrazí uživateli okno s výzvou k zadání uživatelského jména a hesla. Po zadání těchto údajů je klient odešle serveru, kde se tyto údaje ověří (například porovnají s údaji z databáze). Po úspěšném ověření je uživateli povolen přístup. Po správném přihlášení se uživateli zobrazí úvodní stránka. V současné implementaci tato stránka zobrazuje pouze uvítací zprávu, ale jako rozšíření by zde mohl zahrnovat jistý systém upozornění, například na nejbližší události, nebo se zde mohou zobrazovat určité firemní aktuality.

Navigace je řešena pomocí hlavního menu, které se zobrazuje v levé části obrazovky. Protože se jedná o hlavní menu, je viditelné a přístupné z každé stránky celého systému. Celé menu obsahuje bloky s odkazy. V závislosti na tom, jaká má uživatel oprávnění, jsou mu zobrazeny příslušné bloky. Například pokud je uživatel zaměstnancem, ale není vedoucím žádného projektu ani manažerem, zobrazí se mu pouze první blok. V něm jsou odkazy na různé stránky, tyto odkazy odpovídají jednotlivým případům použití z Use Case diagramu. Stejně tak pokud bude uživatel IT manažer, přibude ještě poslední blok.

5.3.5.1 Vzhled

Při návrhu barevného provedení stránek jsem se snažil řídit několika zásadami. Barevné schéma by nemělo být příliš pestré, aby nerušilo a neodrazovalo uživatele, ale zároveň takové, aby celkový vzhled systému působil obyčejně a přirozeně. Proto bylo předefinováno barevné nastavení prakticky všech HTML prvků, které se v systému objevují. Vše samozřejmě za použití kaskádových stylů (viz 4.3.5), aby se dal celý vzhled systému kdykoliv jednoduše změnit. Pomocí kaskádových stylů je řešeno i pozicování jednotlivých elementů, ať už relativní nebo absolutní.

Každý webový IS potřebuje vhodný návrh vzhledu a podpůrných funkcí. Při dodržování jistých pravidel můžeme dosáhnout velmi dobrého výsledku. Řekněme si sedm základních pravidel webdesignu [8].

- To, že něco můžete udělat, ještě neznamena, že byste to udělat měli.

Webové technologie nabízejí mnohé možnosti a nástroje. Přidání určité technologie může často web zpomalit. Není vždy vhodné využití těchto technologií.

- Výjimka potvrzuje pravidlo.
Pro webdesign téměř neexistují absolutní pravidla. Určitá technologie nehodící se na určitý web, viz výše, se může velice hodit na jiný web.
- Soudcem a porotou jsou koncoví uživatelé.
Na webu by se nemělo vyskytovat nic, co by se většině uživatelů nelíbilo nebo čemu by nerozuměli. Každý uživatel má jiný pohled, ale určité základy jsou pro většinu společné.
- Webdesignér by měl vždy usilovat o získání co nejširších znalostí a zkušeností.
Při řešení libovolného problému bude webdesignér vždy ve výhodě, pokud mu bude důkladně rozumět po technické stránce. Je vždy potřeba vzít v potaz technickou stránku věci. Např. rozdíly mezi prohlížeči.
- Nejlepším přístupem je skromnost.
Vždy se najde atraktivnější webdesign. Pokud se nenechá webdesignér zaslepit vlastní pýchou, má spoustu příležitostí, jak se poučit od jiných.
- Není na světě člověk ten, aby se zavděčil lidem všem.
Každý člověk může mít trochu jiný názor a rozdílné preference. Existují však jemné hranice mezi tím, že se váš design bude líbit většině lidí, a tím, že se pokusíte uspokojit všechny, ale ve skutečnosti úplně neuspokojíte nikoho.
- Snažte se udržovat si přehled o specifikacích a standardech.
Webové specifikace a standardy se neustále mění a měnit budou. Webdesignér by ale měl chápat alespoň základní principy nejnovějších technik.

Dále se můžeme zaměřit na určité filozofie jednotlivých webů. Výsledkem těchto filozofií je dosažení co nejefektivnější komunikace s uživatelem a předávání informací. Můžeme rozdělit na tři hlavní body.

- Estetická stránka.
Působí web profesionálním dojmem? Je jeho vzhled konzistentní s identitou, jakou se snaží prezentovat daná společnost či jednotlivec?
- Použitelnost.
Jak rychle a jednoduše může uživatel vyhledat a zpracovat potřebné informace? Jak složité činnosti přitom musí provést?
- Funkčnost.

Správným naprogramováním by měly být zajištěny funkční aspekty webu, například správná činnost formulářů a textů dynamicky doplňovaných z databáze.

Vzhledem k obrovským rozdílům mezi různými uživateli Internetu a ve využívaném hardwaru a softwaru není žádná z uvedených filozofií ideální pro všechny situace. Jestliže webdesignér pochopí jednotlivé filozofie a jejich silné a slabé stránky, může se snadněji rozhodnout, která z nich povede k nejlepšímu naplnění požadavků na konkrétní navrhovaný web.

5.3.5 Ukázka implementace

V této kapitole jsou ukázky implementace v podobě screenshotů.

informační systém	
menu	údaje uživatele xunamed
home	login xunamed
nastavení	heslo 123456
zobrazit managery	jmeno Jan
zobrazit projekty	prijmeni Janota
přidat projekt	telefon 999 999 999
	mail mail@mail.mail
	pozice ředitel
	oddeleni ředitelna
	prava 1

Obrázek č. 5.6 – Ukázka implementace – úvodní obrazovka po přihlášení.

informační systém

menu

home

nastavení

zobrazit managery

přidat manažera

zobrazit projekty

přidat projekt

odhlásit se

	původní hodnota	nová hodnota
ID		id nelze nastavit
oddělení		<input type="text"/>
stav		<input type="text"/>
zadáání		<div style="border: 1px solid #add8e6; height: 60px; width: 100%;"></div>
datum		<input type="text"/>

Obrázek č. 5.7 – Ukázka implementace – přidání projektu.

informační systém

menu

home

nastavení

zobrazit managery

přidat manažera

zobrazit projekty

přidat projekt

odhlásit se

	původní hodnota	nová hodnota
login:	admin	login nelze změnit
heslo:	admin	<input type="text"/>
jmeno:	Jan	<input type="text"/>
prijmeni:	Novak	<input type="text"/>
telefon:	06090888	<input type="text"/>
mail:	novak@novak.com	<input type="text"/>
pozice:	vsechno	<input type="text"/>
oddeleni:	U novaku	<input type="text"/>
prava:	2	IT manager <input type="button" value="v"/> <input type="button" value="x"/>

Obrázek č. 5.8 – Ukázka implementace – změnění atributů uživatele.

informační systém

menu

home

nastavení

zobrazit managery

přidat manažera

zobrazit projekty

přidat projekt

odhlásit se

login	heslo	jméno	příjmení	telefon	email	pozice	oddělení	práva
admin	admin	Jan	Novak	06090888	novak@novak.com	vsechno	U novaku	2
petr	123	petr	reni	777	neco	neco	neco	1

[editovat](#)

[smazat](#)

Obrázek č. 5.9 – Ukázka implementace – zobrazení všech manažerů.

5.3.6 Problémy při implementaci

5.3.6.1 Bezstavovost HTTP

Jedním z problémů, na které jsem během práce na systému narazil, bylo jak se vypořádat z bezstavovostí HTTP protokolu. Protokol HTTP (Hypertext transfer protocol) definuje pravidla pro přenos textu, grafiky, videa a dalších dat přes World Wide Web. Jak jsem již výše zmínil, jedná se o bezstavový protokol. To znamená, že server nemá k dispozici žádnou informaci o předchozích stránkách (dotazech), které si uživatel vyžádal. Taková zjednodušená implementace významnou měrou přispěla k všudypřítomnosti HTTP, bohužel při vývoji komplexních webových aplikací může způsobovat problémy. Následuje příklad.

V tomto informačním systému se neustále zpracovává a zobrazuje množství databázových tabulek. Výstup je vždy stránkovaný. Počet řádků tabulky, které se zobrazí na jedné stránce, je nastaven pomocí globální proměnné `pagesize` a tato hodnota je nastavena na třicet řádků. Pokud databázový dotaz vrátí více než třicet řádků, výstup se rozdělí do několika stránek a pod tabulkou se zobrazí navigace po těchto stránkách. Dále může uživatel výstup seřadit podle kteréhokoliv sloupce v tabulce.

A nyní k vlastnímu problému. Představme si situaci, kdy uživatel prohlíží tabulku projektů, je někde na druhé stránce výstupu, protože projektů je více než třicet a k tomu si nechal výstup seřadit podle loginu vedoucího projektu. Když už konečně našel projekt, který hledal, chce se podívat na seznam zaměstnanců, kteří na projektu pracují a popřípadě provést nějaké změny. Nyní klikne uživatel na odkaz zpět na projekty a bude chtít pokračovat v jejich prohlížení, ale bohužel výstup je opět v takovém stavu jako byl na začátku. To znamená před tím, než ho uživatel seřadil podle loginu vedoucího projektu a než se dostal na druhou stránku výstupu. To může být docela nepříjemné, zejména v případě, kdy uživatel prohlíží jednotlivé projekty a jejich zaměstnance jeden po druhém a po každé provedené změně bude muset znovu seřadit výstup a proklikat se na požadovanou stránku.

Jedním ze způsobů, jak tento problém řešit je neřešit ho a k navigaci používat historii navštívených stránek, která je k dispozici v prohlížeči (používat tlačítko zpět). Tento způsob má několik nevýhod, historie tlačítka zpět není neomezená a navíc se nejedná o pohodlné řešení.

Takže jsem se rozhodl tento problém řešit za použití cookies. Při prvním přístupu na stránku se do cookies uloží název sloupce, podle kterého se bude výstup řadit. Pokud se

uživatel rozhodne seřadit tabulku podle jiného sloupce, přepíše se název sloupce v cookies. To samé platí ohledně stránek výstupu. Takže když uživatel opustí stránku a později se na ni vrátí, zobrazí se mu v takovém stavu, v jakém ji opustil. [6]

5.3.7 Náměty na rozšíření

V této kapitole uvádím pár nápadů, které mě napadly v průběhu práce na systému a tyto nápady by mohly vylepšit a přidat další funkce. Ačkoli již nyní informační systém zvládá řadu funkcí. Ty vycházejí ze základních požadavků na systém, které jsou specifikovány v kapitole 2.

5.3.7.1 Pokročilejší vyhledávání

Uživatel může v systému vyhledávat údaje podle jednotlivých klíčových slov. Například uživatel prohlíží seznam zaměstnanců a v tomto seznamu může hledat podle jména, příjmení, loginu, emailu, telefonního čísla nebo čísla, které používá na firemním chatovacím programu.

Toto jednoduché vyhledávání umožňuje vždy vyhledávat pouze podle sloupců jednotlivých tabulek nebo pohledů, ve kterých vyhledáváme. To má jistá omezení. Bylo by vhodné rozšířit možnosti vyhledávání o tvorbu komplexnějších požadavků. Například by mohl uživatel zadat více klíčových slov zároveň.

5.3.7.2 Použití XML

Extensible Markup Language (XML) patří do kategorie značkovacích jazyků. Tento jazyk se používá i mimo oblast internetu a dovolím si říct, že revoluční, co se týká přenosu dat.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje připojeným stylem. Další možností je pomocí různých stylů provést transformaci do jiného typu dokumentu, nebo do jiné struktury XML [7].

XML také používá tagy jako XHTML. Ale výčet těchto tagů není pevně stanoven. Jde si tedy vytvářet vlastní tagy, které mají uživatelem definovaný význam – je na něm, jak tyto tagy zpracuje. XML dokumenty lze v určitých případech použít i jako zástupce databáze. Může se jednat např. o seznamy knih v eshopu. V souboru je definována struktura popisující knihu — autor, žánr, rok vydání, počet stran, jazyk, atd. Dokumenty XML mají určitě velkou budoucnost, proto informační systémy by měli zahrnovat jejich podporu.

5.3.7.3 Statistika postupu práce

Dalším možným rozšířením by mohla být statistika postupu práce na projektech. Například bylo by výhodné najít vhodný způsob, aby vedoucí mohl být informován o stavu postupu jednotlivých prací.

Například následujícím způsobem. Vedoucímu na základě hodnot od zaměstnanců, kteří by svou hodnotu někam ukládali (nejvhodnějším místem by byl další atribut v tabulce projektů), by se u každého projektu zobrazoval jistý tzv. progress bar, který by informoval, v jakém stavu se projekt (nebo jeho dílčí část) nachází.

5.3.7.4 Rozšíření funkcionality systému o prvky ERP systémů

Možnosti funkcionality ERP systémů jsou velké a rozšíření o některé prvky našeho systému by znamenalo větší použitelnost. Navíc informačním systémem je vyvíjen modulárně, takže informační systém lze lehce o případné požadavky rozšířit. K těmto prvkům resp. požadavkům například patří:

Obecné požadavky

Plná a obousměrná integrace všech modulů do všech částí systému.

Standardní rozhraní pro napojení externích aplikací, které neobsahuje hlavní systém.

Sledování změn dokladů (opravy a mazání)

Využití nástrojů work flow, tj. sledování toku dokumentů v podniku včetně jejich ověřování a schvalování.

Kontroly před hromadným zaúčtováním dokladů

Požadavky na modul výroby

Podpora sériové výroby

Tvorba kusovníků technologických postupů podporující stromovou strukturu

Vizualizace kusovníků a technologických postupů, mapování výrobního procesu a vytváření montážních návodů.

Porovnání nákladů jednotlivých výrobků.

Podpora zajištění materiálových a kapacitních požadavků

Sestavování kapacitních plánů a rozpisů práce na jednotlivá pracoviště.

Vyhodnocování výkonů pracovišť a pracovníků s vazbou na mzdový modul.

Operativní rozhodování o realizaci operací technologického postupu v kooperaci, kontrola nákladů na kooperaci.

Automatické detailní účtování o výrobě (polotovary, nedokončená výroby, hotové výrobky, výdejky, převodky, příjemky)

Evidence neshodných produktů, odchylek a zmetků

Požadavky na modul nákupu

Hromadné generování nových objednávek (dle stavu skladu)

Vystavování jednorázových objednávek bez vazby na sklad

Evidence dodavatelů na skladových kartách

Účtování dokladů

Párování faktur s objednávkami, kontrola a vyčíslení rozdílů

Požadavky na modul skladového hospodářství

Vedení skladové evidence pro libovolné množství skladů a skladových položek.

Podpora čárových kódů EAN a pokladních kódů PLU.

Evidence obrázků na skladových kartách.

Možnost hromadného označení položek ve skladu a jejich jednorázového převzetí do dokladů.

Možnost mazání a opravy skladových dokladů.

Podpora pro provedení inventury na libovolném skladě v libovolný čas.

Požadavky na modul prodeje

Zařazování firem do dealerských skupin

Možnost nastavit individuální ceny, slevy, bonusy, splatnosti a jiné podmínky v závislosti na firmě

Tvorba ceníků

Sledování pohledávek u odběratele (i při nové fakturaci)

Vytváření nabídek

Přijímání objednávek

Rezervace zboží na skladě včetně určení priority

Možnost plného či částečného vyřízení objednávky

Podpora odpisu prodeje ze zásob skladů a automatické zaúčtování prodeje

Vracení položky na sklad

Potvrzování dobropisů s vazbou na zákon o DPH

Podpora danění záloh s vazbou na zákon o DPH

Podpora cizích měn

Vystavování upomínek a penalizačních faktur

Požadavky na modul pokladna a banka

Podpora neomezeného množství pokladen a bankovních účtů v libovolných měnách

Automatický výpočet kurzových rozdílů při uzávěrkách pokladen a účtů v cizích měnách

Podpora agendy platebních příkazů a platebního kalendáře

Možnost obousměrné elektronické výměny dat mezi systémem a bankami

Požadavky na modul majetek

Evidence hmotného a nehmotného dlouhodobého majetku

Členění majetku dle druhu, fyzického umístění, nákladového střediska a zodpovědné osoby

Podpora definovatelných daňových a účetních odpisových skupin

Podpora přecenění a technického zhodnocení

Možnost částečného či úplného vyřazení majetku

Podpora přímé vazby dokladů do účetnictví

Požadavky na modul personalistika

Evidence osobních karet zaměstnanců a hodnocení zaměstnanců

Podpora mezd nad 25 zaměstnanců

Podpora účtování mezd
Podpora nákladových středisek a odpovědných osob
Elektronické podávání výkazů pro správu sociálního zabezpečení, finanční úřady, apod.
Napojení na docházkový systém
Možnost tvorby mzdových složek z výroby
Podpora sociálních výhod (penzijní, životní a kapitálové pojištění)
Podpora pro výběrová řízení
Podpora pro analýzu pracovníků a pracovních míst
Podpora personálního plánování a řízení lidských zdrojů (školení, kontroly, apod.)

Požadavky na modul účetnictví

Podpora účtového rozvrhu s minimálně šestimístnou analytikou
Podpora cizích měn
Účtování interních dokladů
Možnost vzájemných zápočtů
Kurzovní lístek včetně aktualizace dle ČNB (nejlépe automaticky přes internet)
Účtování po střediscích, zakázkách, obchodních případech a zodpovědných osobách
Pevná vazba mezi prvotními doklady a jejich zaúčtováním
Možnost sestavení a přímého tisku přiznání k DPH a souhrnného hlášení s možností elektronického odeslání na finanční úřad

Tyto požadavky jsou příkladem a mohlo bych jich být více a záleží na přesných požadavcích té firmy, pro kterou je nakonec přesně informační systém vyvíjen.

5.4 Ekonomické zhodnocení vlastního řešení

Hrubý odhad nákladů na takovýto systém by mohl být následující:

Tabulka č. 5.1 - Ekonomické zhodnocení vlastního řešení

Krok	Popis	Trvání	Cena
Vytvoření systému	Analýza a naprogramování systému včetně odzkoušení funkčnosti u zákazníka	2-3 měsíců	150 000 Kč
Nákup HW a SW	Pokud není momentální stav dostačující (u firmy Unicode systems je dostačující)	1 měsíc	100 000 – 300 000 Kč (cena může být mnohonásobně vyšší díky např. počtu nově nakoupených počítačů)
Nasazení systému u zákazníka	Instalace serverů, instalace systému, testovací provoz	1 měsíc	20 000 Kč
Náklady na údržbu (paušálně)			20 000 Kč
Zisk z produktu samotného informačního systému			30 000 Kč
Celkem		3 – 4 měsíce	200 000 Kč (v případě IS pro firmu Unicode systems)

Na vývoj a zhotovení systému je třeba dva až tři měsíce a to znamená pro samotného programátora v nákladech cca. 150 000. Co se týká nákupu HW a SW, což ve firmě

Unicode systems není třeba, tak se hlavně odvíjí dle počtu nově nakoupených počítačů. Náklady na údržbu se řeší formou paušálu za menší požadavky. Za větší požadavky typu celých modulů a funkcí se řeší pomocí nové smluvní dohody přímo související se změnou.

Informační systém je vyvíjen jako produkt pro firmu Unicode systems, ale jeho hodnota může být příhodná i pro ostatní firmy. Počáteční zisk z prodeje tohoto systému firmě Unicode systems je jen počátkem prodeje ostatním firmám. Poté je přínos mnohem větší v podobě zisku. Informační systém je z počátku s menším ziskem zaplacen a na trhu zaplňuje díru na trhu a lze očekávat další kupce tohoto systému i v obměněné podobě, protože různorodost a funkcionality systému je velká z důvodu, že je systém vyvíjen modulárně.

Cena takového řešení je tedy poměrně nízká, i když se jedná o pouhý velmi hrubý odhad jak finančních, tak časových zdrojů, existuje možné riziko zpoždění projektu (běžné je o čtvrtinu až třetinu) a tím i navýšení nutných financí. Taktéž pokud by se zvyšovala funkcionality informačního systému, tak i finanční a časová náročnost by se zvýšila.

Jedná se tak o projekt s jistým výsledkem a při nízkých nákladech na jeho realizaci. Pokud ovšem nebude na trhu vhodný (ať již funkčně či finančně) hotový systém, bude to vhodné řešení než se do tohoto projektu pustit.

5.5 Porovnání uvažovaných variant

Srovnáním uvažovaných variant dostáváme následující tabulku, která nám poměrně přesně odhadne, o jakých časových a finančních nákladech bude nutné při změně informačního systému podniku uvažovat.

Tabulka č. 5.2 - Porovnání uvažovaných variant

	Systém na zakázku	ABRA G4	Bílý motýl	IS K2	KARAT Enterprise	QI
Cena v Kč (implementace + první rok)	0,2 milionu	2,85 milionu	3,35 milionu	4,35 milionu	7,35 milionu	6,6 milionu
Náklady na údržbu za 1 rok v Kč	20 000 (paušálně)	225 000	225 000	375 000	750 000	500 000
Doba implementace	3 – 4 měsíce	18 měsíců	6 měsíců	10 – 12 měsíců	6 měsíců	5 měsíců

Do přehledu není zahrnut nutný hardware, který by například u software ABRA znamenal nákup terminálových serverů (v ceně okolo 100 000Kč každý) a výměnu všech pracovních stanic (v současné době probíhá) v ceně okolo 1,25 milionu.

Jaké tedy zvolit řešení? Za předpokladu, že všechny navrhované IS zcela vyhovují požadavkům podniku (tento předpoklad by však bylo nutné ještě ověřit), vychází kombinací nákladů na implementaci a doby implementace nejlépe systém mnou vyvinutý systém na zakázku a IS Bílý motýl. Software ABRA G4 je o něco levnější než IS Bílý motýl, nicméně doba implementace 18 měsíců jej z tohoto pohledu značně znevýhodňuje. Ostatní SW je značně dražší a nepřináší ani úsporu nákladů na údržbu, ani zkrácení doby implementace.

V porovnání nejsou uvažovány další parametry, které by výběr nového IS mohly ovlivnit – zejména není řešena velikost dodavatele a jeho schopnost reagovat na problémy se softwarem během stanovené doby (například při kompletním výpadku systému bude většina dodavatelů schopna zajistit nápravu během dvou dní – zásah během čtyř hodin od nahlášení výpadku jich zvládne značně méně, neboť menší dodavatelé na to nemusí mít dostatečnou kapacitu pracovních sil). Není také řešena

garance dostupnosti, která se může lišit – firma by zřejmě měla požadovat 99% dostupnost služeb či vyšší.

Z tohoto porovnání ještě vyplývá, že vhodnost systému se může lišit podnik od podniku a i když mnou navrhnutý systém je časově a finančně výhodný, tak nemusí být pro všechny malé podniky zcela dostačující.

5.6 Struktura úkolů a činností řešení

Tato kapitola se zabývá úkoly a činnostmi, které by měla firma udělat při zavádění informačního systému do podniku.

1. Rozhodnutí o změně systému, kterou zajistí vedení společnosti. Vedení společnosti by se nemělo bát nových technologií, postupů a inovací.
2. Spojení a konzultaci s dodavateli s otázkou funkcionality a nabídky
3. Výběr vhodného systému. Posouzení nejen ceny ale i vhodnosti a vhodné implementace požadavků na informační systém. Tento výběr by mělo vedení společnosti taktéž konzultovat s vedoucími oddělení.
4. Instalace nového hardwaru. Tento úkol je možné vynechat, pokud dočasný hardware je dostačující.
5. Instalace nového informačního systému
6. Importování starých dat do nového informačního systému. Ulehčí to řadu problémů a času.
7. Testovací provoz nového informačního systému. Například porovnáváním dat se starým systémem
8. Odladění případně zvyklých kritických chyb nového systému
9. Proškolení uživatelů. Důležitý prvek pro snadné zavedení systému do běžného provozu.
10. Přejít do ostrého provozu. Po řádném otestování a proškolení, by měl být přechod na nový systém.
11. Na závěr by bylo vhodné udělat závěrečné zhodnocení výsledků projektu

5.7 Harmonogram a milníky řešení

Tato kapitola se zabývá harmonogramem a milníky řešení nasazení nového informačního systému do podniku z pohledu firmy Unicode systems.

1. Porada týkající se nasazení vhodnosti nového řešení nasazení informačního systému. Oslovení dodavatele informačních systémů s žádostí o nabídku, získání nabídky a popř. vhodnosti funkcionality. Doba trvání cca 1-2 měsíc.

2. Výběr vhodného informačního systému. Doba trvání cca 1 měsíc.

3. Konečný výběr informačního systému.

4. Kontaktování zvoleného dodavatele, sepsání smluv. Doba trvání cca 1 měsíc.

5. Nákup vhodného HW. Je-li nutnost koupě nového HW, tak doba trvání cca 1 měsíc.

6. Implementace informačního systému. Doba trvání cca 3 – 6 měsíců.

7. Nasazení informačního systému do provozu.

8. Testovací provoz. Doba trvání cca 2-4 měsíců.

9. školení uživatelů. Je možné školit uživatele souběžně s testovacím provozem Doba trvání cca 1 – 3 měsíce.

10. Ostrý provoz pod dohledem dodavatele. Doba trvání cca 3 měsíců.

11. Ostrý provoz nového informačního systému.

12. Zhodnocení výsledků zavedení nového informačního systému. Doba trvání cca 1 měsíc.

13. Projekt završen a dokončen.

Celý proces změny informačního systému v podniku tedy by mohl trvat 13 – 22 měsíců. Na doby trvání může mít vliv celá řada faktorů a hodnoty jsou pouze orientační. Milníky jsou označeny tučně.

6 Závěr

Hlavním důvodem, který vede podniky k zavádění nových IS je hlavně větší dostupnost informací, vždy však záleží na konkrétních podmínkách v daném podniku, ale obecně lze očekávat především následující efekty. Vyčíslitelné jako zmenšení stavu zásob, zvýšení produktivity kancelářské práce, zmenšení celkové ceny nákupu materiálu, zlepšení vlastních finančních zdrojů, zmenšení rozpracovanosti a i nevyčíslitelné. Z vnitropodnikového hlediska především rychlé obdržení přesných údajů o stavu podniku - zpráhlednění informačních toků podniku a zlepšení informovanosti o stavu zakázek, zmenšení počtu zastaralých položek, atd. ve vztahu k zákazníkům je to rychlejší a kvalitnější zpracování nabídek, zlepšení pověsti podniku, dodržování termínu a zkrácení průběžných dob. Celkově jsou přínosem vyšší zisky, které jsou dány rovněž schopností pomocí kvalitního systému IS získávat konkurenční výhodu na trhu. I když bude vybrána jakákoliv varianta, tak až po následné implementaci je možné zjistit, jestli bylo dosaženo cílů firmy.

Rozvoj informačních systémů neustále posouvá představy o podnikání. Schopnost neustále rychle reagovat a jednoduše zadávat a získávat informace. To je jeden ze základů úspěšného podnikání v dnešní době. Tento systém umožňuje firmě využívat jeho hlavní výhody a to šetří času. Zvyšuje se i pohodlnost a jednoduchost tím, že systém redukuje často problematickou osobní komunikaci mezi zaměstnanci.

I když tento systém byl v praxi firemního prostředí ověřen jen ve firmě Unicode systems, tak stanovené cíle byly dosaženy. Systém poskytuje jednoduché intuitivní ovládání, které by nemělo způsobovat problémy i méně zručným uživatelům.

6.1 Zhodnocení práce

Na závěr bych se rád zmínil o celkovém přínosu této práce. Díky několika měsíční práci na tomto projektu jsem si zlepšil své dosavadní znalosti z mnoha oblastí, jako například průzkumu mezi zaměstnanci podniku, kalkulace nákladů na informační systém, použití SWOT analýzy, použití skriptovacího jazyka PHP, značkovacího jazyka XHTML nebo práce s databázovým relačním serverem MySQL. Mohu prohlásit, že tato práce pro mě byla opravdovým přínosem a nejen přínosem, o kterém se výše psal. Také tato práce je přínosem pro malé firmy jako Unicode systems zejména v podobě snížení nákladů,

zvýšení zisku apod. Také ziskovost produktu samotného je na dobré úrovni, ale záleží na mnoha okolnostech např. na marketingu a uplatnění na trhu apod.

Literatura

- [1] Kučerová, H. Projektování informačních systémů 2007/2008 [online]. Poslední modifikace: 2007-03-04. [cit. 2008-05-07]. Dostupné na URL: <<http://web.sks.cz/users/ku/pri/usecase.htm>>.
- [2] Jones, M. P. Základy objektově orientovaného návrhu v UML / Vyd. 1. Praha: Grada, 2001. 367 s. ISBN 80-247-0210-X.
- [3] Pavlíček, L. Návrhové vzory [online]. Poslední modifikace: 2005-06-16. [cit. 2008-05-07]. Dostupné na URL: <<http://objekty.vse.cz/Objekty/Vzory-historie>>.
- [4] Pecinovský, R.. Návrhové vzory / Vyd. 1. Brno: Computer Press, 2007. 527s. ISBN 978 0-251-1582-4.
- [5] Kučerová, H. Projektování informačních systémů 2007/2008 [online]. Poslední modifikace: 2007-03-23. [cit. 2008-05-07]. Dostupné na URL: <<http://web.sks.cz/users/ku/PRI/aktivita.htm>>.
- [6] Krčmář, Martin. Informační systém pro řízení projektu / Brno: Vysoké učení technické, 2006. 27 s.
- [7] Soldát, M. Slabikář XML [online]. Poslední modifikace: 2002-05-13. [cit. 2008-05-07]. Dostupné na URL: <<http://interval.cz/serialy/slabikar-xml/>>
- [8] Mlynář, Petr. Informační systém družstva malé kopané / Brno: Vysoké učení technické, 2006. 66 s.
- [9] Gilmore, Jason W. Velká kniha PHP 5 MySQL :kompendium znalostí pro začátečníky i profesionály / Vyd. 1. Brno: Zoner Press, 2005. 711 s. ISBN 80-86815-20-X
- [10] Větrovská, P. Úvod do PHP [online]. Poslední modifikace: 2005-10-08. [cit. 2008-05-07]. Dostupné na URL: <<http://www.webtvorba.cz/php/>>.
- [17] Ponkrác, M. Úvod do databází [online]. Poslední modifikace: 2004-07-29. [cit. 2008-05-07]. Dostupné na URL: <<http://www.zive.cz/h/Programovani/AR.asp?ARI=1176558CA1=2O38>>.
- [18] Kosek, J. Jak pracují databáze na Webu, Co je to databáze [online]. Poslední modifikace: 1999- 02-02. [cit. 2008-05-07]. Dostupné na URL: <<http://www.kosek.cz/clanky/iweb/12.html>>.
- [19] Janovský, D. Úvod do JavaScriptu [online]. Poslední modifikace: 2008-01-14. [cit. 2008-05-07]. Dostupné na URL: <<http://www.jakpsatweb.cz/javascript/javascript-uvod.html>>.

- [20] Pavlíček, L. Návrhové vzory [online]. Poslední modifikace: 2005-06-16. [cit. 2008-05-07]. Dostupné na URL: <<http://objekty.vse.cz/Objekty/Vzory-prehled>>.
- [21] Radovan, L. Úvod do informačních systémů [online]. Poslední modifikace: 2005-06-16. [cit. 2008-05-07]. Dostupné na URL: <<http://radovan.blogger.cz/it/informacni-systemy/uvod-do-informacnich-systemu>>.
- [22] Informační systém [online]. Poslední modifikace: 2010-01-16. [cit. 2010-05-10]. Dostupné na URL: <<http://www.itsolution.cz/informacni-system.a12.html>>.
- [23] Informační systém současného podniku [online]. Poslední modifikace: 2000-04-01. [cit. 2010-05-10]. Dostupné na URL: <<http://seminarky.cz/Informacni-system-soucasneho-podniku-6662>>.
- [24] Basl, J. Současné podnikové informační a řídicí systémy [online]. Poslední modifikace: 2004-07-01. [cit. 2010-05-10]. Dostupné na URL: <<http://www.automatizace.cz/article.php?a=204>>.
- [25] Dodávka, L. Vnitrofiremní komunikace ve vybrané obchodní společnosti / Brno : Masarykova univerzita, 2010. 82 s.
- [26] IT zpravodajství - katalog IT řešení a dodavatelů [online]. © 2001-2009 [cit. 2009-11-08]. Dostupný z WWW: <<http://www.systemonline.cz/>>.
- [27] ABRA Software a.s. - Informační systémy ABRA [online]. © 2009 [cit. 2009-11-08]. Dostupný z WWW: <<http://www.abra.eu/>>.
- [28] BM Servis s.r.o. [online]. [2008] [cit. 2009-11-08]. Dostupný z WWW: <<http://www.bmservis.cz/>>.
- [29] K2 atmitec - Refresh! [online]. © 2006 [cit. 2009-11-08]. Dostupný z WWW: <<http://www.k2atmitec.cz/>>.
- [30] Podnikový informační systém pro každou firmu - QI.cz [online]. 2009 [cit. 2009-11-08]. Dostupný z WWW: <<http://www.qi.cz>>.
- [31] Informační systém KARAT [online]. © 2008 [cit. 2009-11-08]. Dostupný z WWW: <<http://www.karatsoftware.cz/>>.

Seznam obrázků

Obrázek č. 4.1 – Vlivy na správnou implementaci a provozování ERP	23
Obrázek č. 5.1 – Use Case diagram informačního systému pro správu počítačového centra.....	36
Obrázek č. 5.2 – Typy architektur	39
Obrázek č. 5.3 – Diagram tříd navrženého systému	43
Obrázek č. 5.4 – Sekvenční diagram	45
Obrázek č. 5.5 – Diagram aktivit	47
Obrázek č. 5.6 – Ukázka implementace – úvodní obrazovka po přihlášení	56
Obrázek č. 5.7 – Ukázka implementace – přidání projektu.....	57
Obrázek č. 5.8 – Ukázka implementace – změnění atributů uživatele	57
Obrázek č. 5.9 – Ukázka implementace – zobrazení všech manažerů	57

Seznam tabulek

Tabulka č. 4.1 - Informovanost o hodnotách a cílech.....	18
Tabulka č. 4.2 - Nástroje používané k informování o hodnotách a cílech	19
Tabulka č. 4.3 - SWOT analýza výběru nového řešení provozu podnikového IS.....	21
Tabulka č. 5.1 - Ekonomické zhodnocení vlastního řešení	64
Tabulka č. 5.2 - Porovnání uvažovaných variant.....	66

Seznam příloh

Příloha 1. – CD-ROM nosič.